

The Development of MATLAB Based Tools for Antenna Array Analysis



Robey Cecil Beswick

Department of Electrical and Electronic Engineering
University of Stellenbosch

Study leader: Prof. MM Botha

Thesis presented in partial fulfillment of the requirements for the degree of
Master of Engineering in the Faculty of Engineering at Stellenbosch University

M.Eng Electrical and Electronic

March 2020

Dedicated to my Father.

Declaration

I, the undersigned, hereby declare that the work contained in this final year project is my own original work and that I have not previously in its entirety or in part submitted it at any university for a degree.

Robey C. Beswick

.....

Signature

March 2020

.....

Date

Copyright © 2020 Stellenbosch University
All rights reserved

Acknowledgements

I would like to acknowledge the following people for their contribution to this project:

- Prof MM Botha for his constant support and patience throughout the duration of my thesis.
- Wilco Strydom for providing good insight throughout the course of my masters.
- The Stellenbosch E&E department for being the best in the world.
- My Family for the non-stop support throughout this journey.

Abstract

The electric field integral equation's (EFIE) solution using the Rao–Wilton–Glisson (RWG) implementation of the method of moments (MoM) is a complicated process employed in finding the solution of electromagnetic fields numerically. Long and complicated programs must be developed for its custom application. This thesis develops a MoM solver to be used as a basis for custom applications. The solver is developed with custom logic applications to process mesh data, allowing for the solution of selected impedance matrix entries. It employs Gaussian quadrature for standard integration and Radial angular square root methods to deal with singularities. The solver and its toolkit are then employed to test the viability of Windowed macro-basis functions in a three dimensional context.

Contents

Acknowledgement	iii
Abstract	iv
Contents	v
List of Figures	viii
List of Tables	ix
Nomenclature	x
1 Introduction	1
1.1 Project motivation	1
1.2 Outline of thesis	2
2 Formulation of the 3D method of moments	4
2.1 Introduction	4
2.2 Electromagnetic theory	4
2.2.1 Maxwell's equations	4
2.2.2 Boundary conditions for PEC structures	5
2.2.3 General radiating electric field	6
2.2.4 The electric field integral equation	8
2.3 The method of moments	10
2.3.1 A fundamental conceptualisation	10
2.3.2 The Rao–Wilton–Glisson basis function	12
2.3.3 Application of Galerkin testing to the EFIE	14
2.3.4 RWG MoM formulation	16
2.3.5 Conclusion	18
3 Development of a MATLAB based MoM library	20
3.1 The MoM Solver	21
3.1.1 Raw mesh data description	21

CONTENTS

3.1.2	Practical implementation of RWG basis functions	21
3.1.3	Edge finding algorithm	23
3.1.4	Common basis function selection and mesh reduction	26
3.2	Impedance matrix solution	30
3.2.1	Solver design	30
3.2.2	Integration regions	36
3.3	MoM solver results and benchmark testing	40
3.3.1	Convergence of solution	41
3.3.2	Near singularity problems	41
3.3.3	Runtime Analysis	42
3.4	Conclusion	44
4	Application of MoM solver in efficient antenna array analysis	45
4.1	Prominent MBF methods	45
4.2	Linear combination of MBFs at a point	45
4.3	Application of Windowed MBFs to the MoM solution procedure	47
4.4	Enhancement of the Windowed MBF	49
4.4.1	Intuitive approach	49
4.4.2	Defining the Windowed MBF radius	50
4.4.3	Windowed MBF solution method	51
4.5	Results of applying the Windowed MBF method using the MoM solver	53
4.5.1	Primary Windowed MBFs compared to the MoM	53
4.5.2	Windowed MBFs perfect reconstruction of the MoM	55
4.5.3	Windowed MBFs applied to a large scale problem	56
4.5.4	Conclusion	56
5	Conclusions and recommendations	59
	References	62

List of Figures

2.1	An arbitrary PEC surface in free space	9
2.2	A triangular pair sharing the n^{th} edge over which the RWG basis functions are defined	13
2.3	Interaction of an observer and source triangle	18
3.1	Data outputted by <i>importNastran.m</i> or required for custom mesh setup	22
3.2	Illustration of the MoM solver's data structure convention	24
3.3	Small PEC plate with connectivity list	25
3.4	Small PEC plate with connectivity list	27
3.5	New mesh created by <i>BasisFunctionSelect.m</i>	29
3.6	Process of basis function renumbering	30
3.7	MoM solver code representation	31
3.8	Levels of integration accuracy	37
3.9	Integration logic 1	39
3.10	Integration logic 2	40
3.11	\mathbf{Z}_{MoM} convergence to a norm value	42
3.12	\mathbf{Z}_{MoM} norm as parallel plates are moved closer to each other	43
3.13	Time to calculate the impedance matrix for an increasing number of unknowns	44
4.1	Current distribution over an arbitrary antenna	46
4.2	Arbitrary current distribution over an array of bowtie elements	47
4.3	Multiple bowtie \mathbf{I} vector illustration	50
4.4	Solution radius 1, 2 and 3 illustration	51
4.5	Solution radius 1 and 2 illustration	52
4.6	Difference of the primary case Windowed MBF approximation from the MoM at varying distances between Array elements	54
4.7	Scattered farfield of a 4x4 bowtie array in the XY plane	55
4.8	Difference of varying radius Windowed MBF current vector solutions from MoM solution	56

LIST OF FIGURES

4.9	64 Bowtie scattered field in XY plane	57
4.10	64 Bowtie scattered field in XY plane difference from MoM solution . .	57
4.11	Difference of varying radius Widowed MBF I vector solutions from full MoM solution of 64 bowties	58

List of Tables

3.1	Selectable integration modes	38
3.2	Integration points used to test convergence	41
3.3	Difference between FEKO and MoM solver matrix norms at near singu- larity points	42
3.4	MoM solver runtime for increasing amounts of unknowns	43

Nomenclature

Acronyms

2D	Two dimensional
ACA	Adaptive cross approximation
CBF	Characteristic basis function
CBFM	Characteristic basis function method
CEM	Computational electromagnetics
GUI	Graphical user interface
MBF	Macro-basis function
MeX	MATLAB EXECUTABLE
PEC	Perfect electric conductor
RAR	<i>Radial – Angular – R^1 – Sqrt</i>
RWG	Rao–Wilton–Glisson
SKA	Square kilometre array
TM	Transverse magnetic

Greek Symbols

ϵ	permittivity
η	wave impedance
μ	permeability

Other Symbols

j	imaginary number
k	wave number

Chapter 1

Introduction

1.1 Project motivation

Since the formulation of Maxwell's equations Electromagnetism has become one of the most widely studied fields of physical science.

Due to the complexity associated with solving Maxwell's equations analytically engineers must often rely on numerical methods to describe the behaviour of large scale electromagnetic problems. Computational electromagnetics often uses surface equivalence models to approximate the behaviour of EM fields due to some structure in space. The approximation generally follows the same order, a mesh is generated to represent the structures surface, assumptions are made as to the physical properties of the structure and a linear system of equations is constructed to describe the current behaviour on the structure.

One of the most prominent methods in modern day computational electromagnetics is the method of moments, an integral equation method that populates a solvable linear system by transforming the governing equations of a boundary-value problem [1]. The MoM's application on the EFIE is particularly effective in modelling the behaviour of currents on the surfaces of open and closed PEC structures [2]. The EFIE MoM is extremely popular as most large scale scatters can be considered PEC structures. This piece focuses on the EFIE MoM method.

The MoM solution yields a dense impedance matrix as a result of its integration kernel, the Green's function [2]. The dense impedance matrix results in a solution complexity of $O(N^2)$ where N is the number of unknowns, the solution can become extremely cumbersome for large scale scattering structures. Due to the MoM's extreme computational complexity many optimization techniques have been developed to enhance its efficiency.

As the prominence of the MoM solution method has increased a multitude of optimizations have been developed for its optimization. The Stellenbosch University CEM

research group extensively studies the MoM solution method and its various optimization techniques. In order to investigate the efficiency of these techniques they must be applied to a three dimensional MoM procedure. The RWG MoM solution procedure is the standard in most commercial and public MoM solvers. When researchers want to study optimization methods such as the ACA [3] and CBFM [4] or their combination [5] they often require special implementations of an RWG MoM solver. Thus, when most students in the CEM group start their post graduate studies they must first implement a custom RWG MoM solver which often results in a rudimentary, low accuracy and low efficiency solver that serves a single purpose and is discarded after the research is completed.

Prof MM Botha noticed the need for an easily customisable, highly accurate, relatively efficient and easy to understand RWG MoM solver implementation that could serve as a basis for the research group to use in their respective fields.

The goal of this project is the design of a powerful MoM solver that fulfils the aforementioned criteria. As the main program used within the Stellenbosch CEM group is MATLAB the choice was made to implement the impedance matrix solution as a MATLAB EXECUTABLE or **MeX**. The MATLAB programming language was developed in the early 2000's, originally in FORTRAN and then eventually C. A **MeX** allows a user to create custom C-based code and compile it into an executable to be treated as a normal MATLAB function. This provides the massive speed advantage of a lower level programming language as well as custom memory management.

This piece serves to detail the process of designing and building this high level MoM code and showcase its power by developing and testing the Windowed MBF method in a three dimensional context.

1.2 Outline of thesis

Chapter 2: A review of the relevant electromagnetic and mathematical theory required for the development of the MoM solver. It starts with Maxwell's equation and highlights the process of developing them into the EFIE after which the MoM's application to the EFIE via RWG basis functions is stipulated. Finally Gaussian quadrature for triangular domains and singularity cancellation quadrature are reviewed.

CHAPTER 1. INTRODUCTION

Chapter 3: The main chapter of this piece first details the intricacies of developing the MoM solver after which the accuracy is showcased and compared to FEKO.

Chapter 4: The MoM solver's effectiveness is showcased by developing and testing the Windowed MBF method in a three dimensional context.

Chapter 5: The work completed during this piece is concluded and future possible expansions of the MoM code are discussed.

Chapter 2

Formulation of the 3D method of moments

2.1 Introduction

The ultimate goal of this project is the design and implementation of an efficient MoM solver for application in various scattering problems. This chapter presents the theoretical knowledge required for the projects implementation. The differential and integral forms of Maxwell's equations are introduced after which they are reviewed in the context of general radiating electromagnetic fields. The EFIE is introduced by applying boundary conditions to the aforementioned electromagnetic field equations.

As the fundamentals required to understand the MoM will have been reviewed the concept is introduced through an abstract conceptualisation and applied to the EFIE. At this point an unsolvable integral equation exists. Basis functions and Galerkin testing are then introduced as a means to solve the equation followed by the classic RWG MoM solution. Finally, Gaussian and singularity cancellation quadrature are introduced as they are crucial to the accurate solution of the RWG MoM.

2.2 Electromagnetic theory

2.2.1 Maxwell's equations

Maxwell's equations form the basis of almost all modern electromagnetic theory. Their establishment in the mid 19th century [6] was one of the most important occurrences in modern history. They are a set of partial differential equations that describe the relations and behaviour of electric and magnetic fields, currents and charges [7]. The appropriate formulation of Maxwell's equations combined with the use of boundary conditions can effectively model almost any electromagnetic problem.

The differential form of Maxwell's equations are most commonly used for the solution of boundary value problems [7]. Given a homogeneous media with permittivity

CHAPTER 2. FORMULATION OF THE 3D METHOD OF MOMENTS

ε and permeability μ , the electric and magnetic fields existing in the medium must satisfy Maxwell's equations [8]

$$\nabla \times \mathbf{E} = -\mathbf{M} - j\omega\mu\mathbf{H} \quad (2.1)$$

$$\nabla \times \mathbf{H} = \mathbf{J} + j\omega\varepsilon\mathbf{E} \quad (2.2)$$

$$\nabla \cdot \mathbf{D} = q_e \quad (2.3)$$

$$\nabla \cdot \mathbf{B} = q_m \quad (2.4)$$

The equations are given in phasor form as wave behaviour is assumed to be *time harmonic*. The remainder of this text uses the assumption that all waves are of cosinusoidal form represented by $e^{j\omega t}$ which is suppressed in the phasor form. \mathbf{M} and q_m are impressed magnetic sources, not physically realisable but used as mathematical tools to solve radiating and scattering problems. The electric and magnetic fields can be related to their respective flux densities as,

$$\mathbf{D} = \varepsilon\mathbf{E} \quad (2.5)$$

$$\mathbf{B} = \mu\mathbf{H} \quad (2.6)$$

The differential forms of Maxwell's equations are coupled, making them difficult to solve. A tool in the effective solution of Maxwell's equations is the use of boundary conditions for special cases of geometry or materials.

2.2.2 Boundary conditions for PEC structures

Vector fields described by Maxwell's equations in differential form must be single valued and continuous. A boundary existing between two media creates a discontinuity in electromagnetic field behaviour causing the differential operators applied to these field vectors to have no meaning at a discontinuity [7]. Thus, conditions for handling discontinuities at boundaries between media must be found.

As this piece deals exclusively with the EFIE MoM formulation only the boundary conditions between free space and PEC structures are presented as [8],

$$\hat{n} \times \mathbf{E} = 0 \quad (2.7)$$

$$\hat{n} \times \mathbf{H} = \mathbf{J}_s \quad (2.8)$$

$$\hat{n} \cdot \mathbf{D} = q_e \quad (2.9)$$

$$\hat{n} \cdot \mathbf{B} = 0 \quad (2.10)$$

 CHAPTER 2. FORMULATION OF THE 3D METHOD OF MOMENTS

Where \hat{n} is the unit normal at the interface and \mathbf{J}_s is the surface current over it. Equation 2.7 states the E-field component tangential to the interface is zero. Later it will become more apparent that this is fundamental to the EFIE MoM formulation.

Maxwell's equations are subsequently manipulated for implementation in solving for radiating and scattering problems. These boundary conditions will be used to simplify the resulting expressions.

2.2.3 General radiating electric field

The general expression for radiating fields due to an electric current distribution is subsequently presented. The classical approach to solving radiation and scattering problems is to substitute equations 2.1 and 2.2 into each other, raising their order and substituting in the equation of continuity[9] yielding a direct solution for the fields. However, it is often extremely complicated or impossible to directly solve for the radiated fields due to a current distribution. To aid in solving for the \mathbf{E} and \mathbf{H} fields, auxiliary vector potentials are introduced [7]. They require more steps but are a simpler approach.

Maxwell's equations in source free, homogeneous media with permittivity ε , and permeability μ are taken as a starting point in using vector potentials to find a general expression for a radiated field due to an electric current

$$\nabla \times \mathbf{E} = -j\omega\mu\mathbf{H} \quad (2.11)$$

$$\nabla \times \mathbf{H} = j\omega\varepsilon\mathbf{E} + \mathbf{J} \quad (2.12)$$

First an expression for the magnetic vector potential is introduced. Since the magnetic field \mathbf{H} is always solenoidal it can be written as the curl of an arbitrary vector, conventionally \mathbf{A} . Thus it is written

$$\mathbf{H} = \frac{1}{\mu} \nabla \times \mathbf{A} \quad (2.13)$$

substituting this into (2.11) yields

$$\nabla \times \mathbf{E} = -j\omega \nabla \times \mathbf{A} \quad (2.14)$$

it is then rewritten as

$$\nabla \times (\mathbf{E} + j\omega\mathbf{A}) = 0 \quad (2.15)$$

 CHAPTER 2. FORMULATION OF THE 3D METHOD OF MOMENTS

Noting that any curl free vector can be written as the gradient of a scalar and the gradient of a scalar is zero (2.15) is equated to a newly introduced *scalar electric potential* Φ_e .

$$\nabla \times (\mathbf{E} + j\omega\mathbf{A}) = 0 = \nabla \times (-\nabla\Phi_e) \quad (2.16)$$

rewriting the above equation for \mathbf{E} yields

$$\mathbf{E} = -j\omega\mathbf{A} - \nabla\Phi_e \quad (2.17)$$

Equation (2.17) is an expression of the radiated field in terms of the vector and scalar magnetic potentials.

It is now illustrated how these vector potentials conform to the Helmholtz equation. We make use of the vector identity

$$\nabla \times \nabla \times \mathbf{A} = \nabla (\nabla \cdot \mathbf{A}) - \nabla^2 \mathbf{A} \quad (2.18)$$

The curl of (2.13) is taken yielding

$$\mu \nabla \times \mathbf{H} = \nabla (\nabla \cdot \mathbf{A}) - \nabla^2 \mathbf{A} \quad (2.19)$$

Now substituting the right side of equation (2.12) for the curl of the magnetic field,

$$\mu \mathbf{J} + j\omega\mu\varepsilon\mathbf{E} = \nabla (\nabla \cdot \mathbf{A}) - \nabla^2 \mathbf{A} \quad (2.20)$$

The expression derived for the radiated field, (2.17), can be substituted into the above equation

$$\mu \mathbf{J} + j\omega\mu\varepsilon(-j\omega\mathbf{A} - \nabla\Phi_e) = \nabla (\nabla \cdot \mathbf{A}) - \nabla^2 \mathbf{A} \quad (2.21)$$

Noting that the wave number $k = \omega\sqrt{\mu\varepsilon}$

$$\nabla^2 \mathbf{A} - k^2 \mathbf{A} = -\mu \mathbf{J} + \nabla (\nabla \cdot \mathbf{A} + j\omega\mu\varepsilon\Phi_e) \quad (2.22)$$

Equation (2.22) is now an inhomogeneous equation with a near impossible solution. The curl of \mathbf{A} is defined in (2.11), the divergence has not yet been defined. It can be set to best suite the solution, as long as it remains consistent throughout the application. The freedom to set the divergence of \mathbf{A} is what makes the auxiliary vector potentials so attractive [8].

The divergence of \mathbf{A} is defined as

$$\nabla \cdot \mathbf{A} = -j\omega\mu\varepsilon\Phi_e \quad (2.23)$$

 CHAPTER 2. FORMULATION OF THE 3D METHOD OF MOMENTS

Which is known as the Lorenz gauge [7] allowing simplification of (2.22) to the inhomogeneous vector Helmholtz equation

$$\nabla^2 \mathbf{A} - k^2 \mathbf{A} = -\mu \mathbf{J} \quad (2.24)$$

The electric field everywhere in a source free region can now be obtained. Substituting (2.23) into the expression of \mathbf{E} in terms of the auxiliary potentials defined in (2.17) yields the final expression for the electric field.

$$\mathbf{E} = -j\omega \mathbf{A} - \frac{j}{\omega\mu\epsilon} \nabla (\nabla \cdot \mathbf{A}) \quad (2.25)$$

Where \mathbf{A} is written in terms of the scalar Green's function $G(\mathbf{r}, \mathbf{r}')$ convolve with the current \mathbf{J} .

$$\mathbf{A}(\mathbf{r}) = \mu \iiint_V G(\mathbf{r}, \mathbf{r}') \mathbf{J}(\mathbf{r}') dv' = \mu \iiint_V \mathbf{J}(\mathbf{r}') \frac{e^{-jk|\mathbf{r}-\mathbf{r}'|}}{|\mathbf{r}-\mathbf{r}'|} dv' \quad (2.26)$$

Equation (2.25) is a general expression for the electric field due to a volumetric current. The current \mathbf{J} must be integrated to solve for the field \mathbf{E} . Writing this out in full yields,

$$\mathbf{E} = -j\omega\mu \iiint_V \mathbf{J}(\mathbf{r}') G(\mathbf{r}, \mathbf{r}') dv' + \frac{1}{k^2} \nabla \iiint_V \nabla' \cdot \mathbf{J}(\mathbf{r}') G(\mathbf{r}, \mathbf{r}') dv' \quad (2.27)$$

The boundary conditions stipulated in section 2.2.2 and surface equivalence principles can now be applied to (2.27) to derive the EFIE.

2.2.4 The electric field integral equation

The preceding section introduced an equation to calculate the general electric field due to a volumetric current \mathbf{J} . The EFIE is based on boundary conditions specific to PEC structures and free space [7]. Recalling from equation (2.7), at the interface between a PEC surface and free space the tangential component of the electric field is zero. Given a PEC surface is placed in the presence of an incident electric field, illustrated in figure 2.1, the fields can be expressed as,

$$\mathbf{E}(\mathbf{r}) = \mathbf{E}_{incident}(\mathbf{r}) + \mathbf{E}_{scattered}(\mathbf{r}) \quad (2.28)$$

Given \mathbf{r} is on the PEC structure's surface, the boundary condition is applied as

$$[\mathbf{E}(\mathbf{r})]_t = [\mathbf{E}_{incident}(\mathbf{r})]_t + [\mathbf{E}_{scattered}(\mathbf{r})]_t = 0 \quad (2.29)$$

$$[\mathbf{E}_{scattered}(\mathbf{r})]_t = -[\mathbf{E}_{incident}(\mathbf{r})]_t \quad (2.30)$$

CHAPTER 2. FORMULATION OF THE 3D METHOD OF MOMENTS

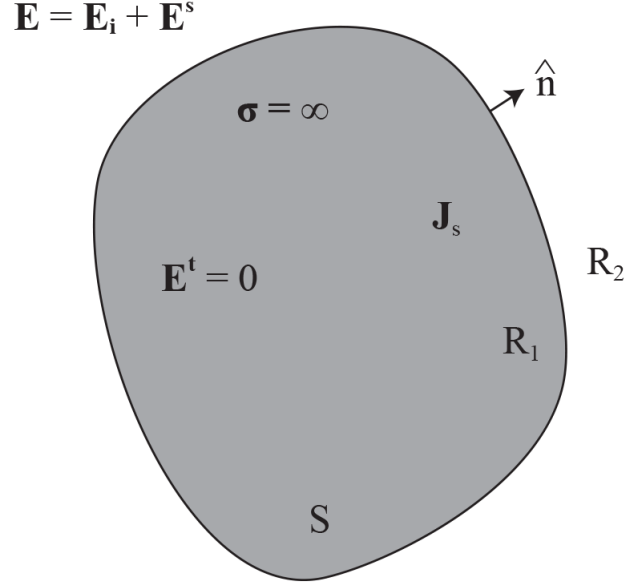


Figure 2.1: An arbitrary PEC surface in free space

The subscript t indicates the field component tangential to the surface S . Observing equation 2.30 the tangential component of the scattered and incident fields are equivalent but opposite in sign on the conductor surface.

The incident field thus induces a surface current yielding a scattered electric field. Given the surface current density \mathbf{J}_s is known, the scattered field can be calculated. Recalling the solution for a general radiated field from equation (2.27) it can simply be modified for a surface current density \mathbf{J}_s as

$$\mathbf{E}_s(\mathbf{r}) = -j\omega\mu \iint_S \mathbf{J}_s(\mathbf{r}') G(\mathbf{r}, \mathbf{r}') ds' + \frac{1}{k^2} \nabla \iint_S \nabla' \cdot \mathbf{J}_s(\mathbf{r}') G(\mathbf{r}, \mathbf{r}') ds' \quad (2.31)$$

It is important to note the ∇ and ∇' operators take the gradient with respect to the observation and source coordinates respectively [7].

Given that \mathbf{r} is restricted to the surface S only, equation (2.30) can be rewritten as,

$$-j\omega\mu \left[\iint_S \mathbf{J}_s(\mathbf{r}') G(\mathbf{r}, \mathbf{r}') ds' + \frac{1}{k^2} \nabla \iint_S \nabla' \cdot \mathbf{J}_s(\mathbf{r}') G(\mathbf{r}, \mathbf{r}') ds' \right]_t = -[\mathbf{E}_i]_t \quad (2.32)$$

CHAPTER 2. FORMULATION OF THE 3D METHOD OF MOMENTS

which is the electric field integral equation. Given the incident field \mathbf{E}_i is known, it is possible to solve for \mathbf{J}_s at any point on the scatterer that can subsequently be used to solve for \mathbf{E}_s .

In practicality most large scale antennas and scatters can be assumed to be PEC structures and the EFIE can be used to solve for their surface currents and scattered fields. Applying the MoM to the EFIE allows for its solution using digital means. The MoM's application to the EFIE yields a solvable matrix equation.

2.3 The method of moments

The MoM is one of the most prominent numerical techniques used in modern day electromagnetics. It transforms the governing equation of a boundary-value problem into a matrix equation, allowing for a solution using digital means. This section is a review of the MoM solution procedure and its application to the EFIE.

2.3.1 A fundamental conceptualisation

In general, problems solved using the MoM follow a similar progression. An abstract example shall be employed to garner an understanding of this progression before it is applied to the EFIE. This description follows that of [6].

We express an integral equation as

$$\mathcal{L}\varphi = \mathcal{F} \quad (2.33)$$

Observing (2.33) \mathcal{L} denotes an integral operator, φ is the solution sought using the MoM and \mathcal{F} is the driving term that is known. In this form the equation is unsolvable. In order to find a solution for φ a finite set of basis functions are chosen as an approximation of the solution

$$\varphi = \sum_{n=1}^N c_n v_n \quad (2.34)$$

where in (2.34) c_n and v_n are the unknown weighting coefficients and known basis functions respectively. Now substituting into (2.33) yields

$$\sum_{n=1}^N c_n \mathcal{L}(v_n) = \mathcal{F} \quad (2.35)$$

CHAPTER 2. FORMULATION OF THE 3D METHOD OF MOMENTS

The substitution yields a sum of coefficients multiplied by a set of corresponding, known, integrals. The fundamental step is converting (2.35) into a matrix equation. The conversion is achieved by multiplying (2.35) by a set of testing functions $w_m (m = 1, 2, 3, \dots, N)$ and integrating over the solution domain Ω [6].

$$\sum_{n=1}^N c_n \int_{\Omega} w_m \mathcal{L}(v_n) d\Omega = \int_{\Omega} w_m \mathcal{F} d\Omega \quad m = 1, 2, 3, \dots, N \quad (2.36)$$

For the sake of understanding, (2.36) is rewritten using $\langle \cdot \rangle$ to represent the integration over the solution domain.

$$\sum_{n=1}^N c_n \langle w_m, \mathcal{L}(v_n) \rangle = \langle w_m, \mathcal{F} \rangle \quad m = 1, 2, 3, \dots, N \quad (2.37)$$

Equation (2.37) represents a set of N linear sums that sum to N , it can be written as a matrix equation

$$[A]\{c\} = \{b\} \quad (2.38)$$

where A is the system matrix, the elements of which are written

$$A_{mn} = \langle w_m, \mathcal{L}(v_n) \rangle \quad m, n = 1, 2, 3, \dots, N \quad (2.39)$$

$\{b\}$ and $\{c\}$ are known as the source and unknown vectors respectively, with the elements of $\{b\}$ given by

$$b_m = \langle w_m, \mathcal{F} \rangle \quad (2.40)$$

Thus, the original equation (2.33) is discretized and solvable. The procedure described in section 2.3.1 is the Method of Moments. A powerful advantage of the MoM is that unlike the finite element method [10] which uses differentiation, the MoM makes use of an integral operator that is more forgiving when choosing basis functions. The finite element method (FEM) requires a differentiation, so the basis functions selected when using the FEM must be at least first order (for their derivative to be non-zero). As the MoM is an integral method zeroth order basis and testing functions can be chosen, such as pulse functions, the choice of these simpler functions can greatly simplify the integrations. The correct choice of basis function is crucial as it must sufficiently describe the field behaviour and be solvable. The Rao–Wilton–Glisson Basis function is subsequently introduced as the choice basis function for this piece.

 CHAPTER 2. FORMULATION OF THE 3D METHOD OF MOMENTS

2.3.2 The Rao–Wilton–Glisson basis function

The previous section introduced the MoM as a means of transforming the governing equation of a boundary–value problem into a matrix equation. The first step taken by the procedure is assigning a finite set of basis functions to approximate a continuous vector as in (2.34). This section introduces the RWG basis function, the most widely used basis function in 3D MoM problems.

For convenience the EFIE is recalled as

$$\left[\iint_S \mathbf{J}_s(\mathbf{r}') G(\mathbf{r}, \mathbf{r}') ds' + \frac{1}{k^2} \nabla \iint_S \nabla' \cdot \mathbf{J}_s(\mathbf{r}') G(\mathbf{r}, \mathbf{r}') ds' \right]_t = \frac{-j}{\omega \mu} [\mathbf{E}_i]_t \quad (2.41)$$

The equation allows one to solve for \mathbf{J}_s anywhere on the PEC surface. Employing the MoM solution will result in a matrix equation that approximates \mathbf{J}_s at N unique points on a surface S . The process's first step is assigning a set of linearly independent basis functions to approximate the surface current. The Rao–Wilton–Glisson method approximates \mathbf{J}_s as $\tilde{\mathbf{J}}_s$ which is defined as,

$$\tilde{\mathbf{J}}_s = \sum_{n=1}^N I_n \mathbf{f}_n(\mathbf{r}) \cong \mathbf{J}_s \quad (2.42)$$

The approximation is a linear combination of independent basis functions. I_n is a set of N coefficients that scale the corresponding basis functions $\mathbf{f}_n(\mathbf{r})$. I_n can be interpreted as the normal component of current density flowing over the n^{th} edge. In order to make this approximation the basis functions cannot include any line or point charges [11]. Rao–Wilton–Glisson basis functions are defined on N interior edges of a triangular surface approximation and only ever exist over two triangles. Figure 2.2 illustrates how a basis function is defined over the n^{th} interior edge of an arbitrary surface S . Each basis function exists only on the triangular pair T_n^+ and T_n^- [11] and vanishes everywhere else on surface S . Any point on the triangular pair can be characterized by \mathbf{r} , alternatively it can be characterized by the local position vector $\boldsymbol{\rho}_n^\pm(\mathbf{r})$ which is defined in terms of T_n^\pm 's free vertex and a point on it as

$$\boldsymbol{\rho}_n^+(\mathbf{r}) = \mathbf{r} - \mathbf{v}^+ \quad \text{in } T_n^+ \quad (2.43)$$

$$\boldsymbol{\rho}_n^-(\mathbf{r}) = \mathbf{r} - \mathbf{v}^- \quad \text{in } T_n^- \quad (2.44)$$

The positive and negative signs assigned to each triangular pair define the choice of current reference direction across the n^{th} edge of the triangle. This also determines

CHAPTER 2. FORMULATION OF THE 3D METHOD OF MOMENTS

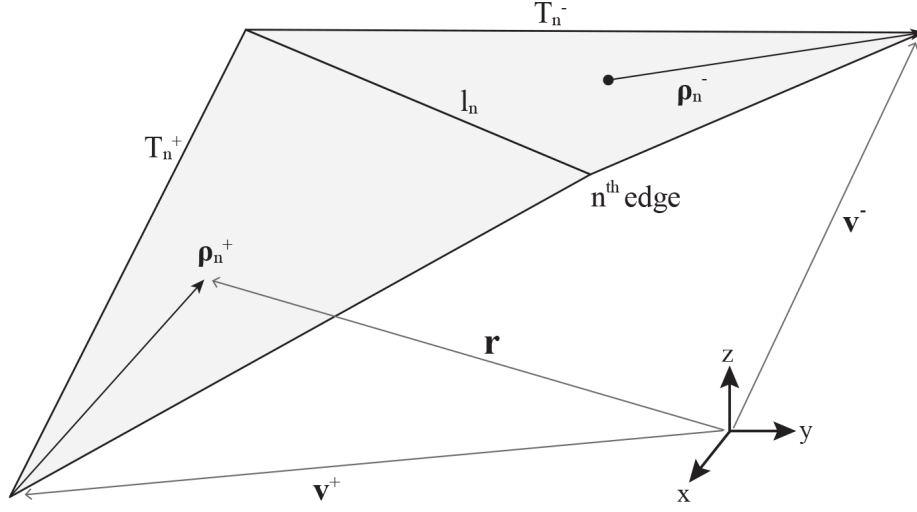


Figure 2.2: A triangular pair sharing the n^{th} edge over which the RWG basis functions are defined

whether the ρ -vector points to or away from the free vertex [11]. Given the aforementioned constraints and parameters the Basis functions are now formally defined as,

$$\mathbf{f}_n = \begin{cases} \frac{l_n}{2A_n^+} \rho_n^+ & \mathbf{r} \text{ in } T_n^+ \\ \frac{l_n}{2A_n^-} \rho_n^- & \mathbf{r} \text{ in } T_n^- \\ 0 & \text{otherwise} \end{cases}$$

where l_n is the length of the n^{th} edge and A_n^\pm are the triangles corresponding areas. These basis functions are specifically chosen to approximate the surface current as they possess a set of properties uniquely suited to this role [11]. They are as follows

1. The current has no component normal to the non-common edges. This ensures no line charges can exist on these boundaries.
2. The component of current normal to the common (n^{th}) edge is constant and continuous [11] across the edge. This is ensured by the normalising factors present in (2.3.2).

 CHAPTER 2. FORMULATION OF THE 3D METHOD OF MOMENTS

3. If one takes the surface divergence of (2.3.2), as will be required in the EFIE (see equation (2.31)) it results in constant values,

$$\nabla_s \cdot \mathbf{f}_n(\mathbf{r}) = \begin{cases} -\frac{l_n}{A_n^+} & \mathbf{r} \text{ in } T_n^+ \\ \frac{l_n}{A_n^-} & \mathbf{r} \text{ in } T_n^- \\ 0 & \text{otherwise} \end{cases} \quad (2.45)$$

Due to the equation of continuity the current divergence is proportional to the surface charge density. Observing equation (2.45) T_n^+ and T_n^- are equal and opposite in sign. This indicates a net charge of zero for the triangular pair.

These properties make RWG basis functions extremely attractive in the MoM's application to the EFIE. Substituting (2.42) into (2.32) yields,

$$\left[\sum_{n=1}^N I_n \iint_{T_n} \mathbf{f}_n(\mathbf{r}') G(\mathbf{r}, \mathbf{r}') ds' + \frac{1}{k^2} \sum_{n=1}^N I_n \nabla \iint_{T_n} \nabla' \cdot \mathbf{f}_n(\mathbf{r}') G(\mathbf{r}, \mathbf{r}') ds' \right] = \frac{-j}{\omega \mu} [\mathbf{E}_i] \quad (2.46)$$

where recalling from section 2.3.1 the EFIE is now a sum of coefficients multiplied by a set of corresponding integrals. All that now remains is to apply a set of testing functions using an inner product [7].

This section introduced RWG basis functions as a means of approximating surface current as a set of N weighted discrete basis functions on a PEC surface. The subsequent section describes Galerkin testing and how its application to (2.46) is the final step in attaining the EFIE MoM solution.

2.3.3 Application of Galerkin testing to the EFIE

The preceding section introduced and employed RWG basis functions to discretize the surface current of a PEC structure. According to section 2.3.1 the application of a testing procedure to the discrete EFIE is all that remains in attaining a solvable matrix equation.

Testing or weighting function's are applied to a boundary equation of via the symmetric inner product [7] defined as,

$$\langle \mathbf{w}, \mathbf{f} \rangle = \iint_S \mathbf{w} \cdot \mathbf{f} ds, \quad (2.47)$$

 CHAPTER 2. FORMULATION OF THE 3D METHOD OF MOMENTS

where \mathbf{w} represents weighting functions on the surface S [7]. Recalling equation (2.36) a set of N testing functions can be applied over the solution domain via the inner product. Equation (2.37) is repeated here for emphasis,

$$\sum_{n=1}^N c_n \langle w_m, \mathcal{L}(v_n) \rangle = \langle w_m, \mathcal{F} \rangle \quad m = 1, 2, 3, \dots, N$$

where the inner product is taken between the weighting coefficients w_m and $\mathcal{L}(v_n)$ (the integral applied to the basis function operator). This procedure can be used in applying testing functions to the discretized EFIE. However, the choice of testing function is crucial in obtaining an easily solvable solution. The testing functions must be linearly independent for the resulting equations to have the same property. The solution complexity is also strongly dependent on the choice of testing function. Galerkin testing, which chooses the basis and testing functions to be the same [7], is used in most MoM applications on the EFIE as it is often forgiving in terms of computational complexity.

In applying Galerkin testing to the EFIE, testing functions are chosen as \mathbf{f}_m where $m = 1, 2, 3, \dots, N$ defined in (2.3.2). They are applied to the EFIE. Choosing the m^{th} testing function the resulting equation is

$$\begin{aligned} \sum_{n=1}^N I_n \left[\iint_{T_m} \mathbf{f}_m(\mathbf{r}) \cdot \iint_{T_n} \mathbf{f}_n(\mathbf{r}') G(\mathbf{r}, \mathbf{r}') ds' ds \right] \\ + \sum_{n=1}^N I_n \left[\frac{1}{k^2} \iint_{T_m} \mathbf{f}_m(\mathbf{r}) \cdot \left[\nabla \iint_{T_n} \nabla' \cdot \mathbf{f}_n(\mathbf{r}') G(\mathbf{r}, \mathbf{r}') ds' \right] ds \right] = \\ \frac{-j}{\omega \mu} \iint_{T_m} \mathbf{f}_m(\mathbf{r}) \cdot \mathbf{E}_i(\mathbf{r}) ds \end{aligned} \quad (2.48)$$

The second term on the left side of (2.48) requires special treatment as the del operator acts on the greens function and will yield a singularity. The dell operator must be moved from the greens function, given the term as

$$\iint_{T_m} \mathbf{f}_m(\mathbf{r}) \cdot \left[\nabla \iint_{T_n} \nabla' \cdot \mathbf{f}_n(\mathbf{r}') G(\mathbf{r}, \mathbf{r}') ds' \right] ds, \quad (2.49)$$

and given the following vector identity,

$$\mathbf{g}(\mathbf{r}) \cdot \nabla \mathbf{h}(\mathbf{r}) = \nabla \cdot [\mathbf{g}(\mathbf{r}) \mathbf{h}(\mathbf{r})] - [\nabla \cdot \mathbf{g}(\mathbf{r})] \mathbf{h}(\mathbf{r}) \quad (2.50)$$

 CHAPTER 2. FORMULATION OF THE 3D METHOD OF MOMENTS

The term in (2.49) can transformed as,

$$\begin{aligned}
 & \iint_{T_m} \mathbf{f}_m(\mathbf{r}) \cdot \left[\nabla \iint_{T_n} \nabla' \cdot \mathbf{f}_n(\mathbf{r}') G(\mathbf{r}, \mathbf{r}') ds' \right] ds \\
 &= \iint_{T_m} \nabla \cdot \left[\mathbf{f}_m(\mathbf{r}) \left(\iint_{T_n} \nabla' \cdot \mathbf{f}_n(\mathbf{r}') G(\mathbf{r}, \mathbf{r}') ds' \right) \right] ds \\
 &\quad - \iint_{T_m} \left[\nabla \cdot \mathbf{f}_m(\mathbf{r}) \left(\iint_{T_n} \nabla' \cdot \mathbf{f}_n(\mathbf{r}') G(\mathbf{r}, \mathbf{r}') ds' \right) \right] ds, \quad (2.51)
 \end{aligned}$$

the first term on the right hand side of (2.51) can be modified using the 2D divergence theorem,

$$\begin{aligned}
 & \iint_{T_m} \nabla \cdot \left[\mathbf{f}_m(\mathbf{r}) \iint_{T_n} \nabla' \cdot \mathbf{f}_n(\mathbf{r}') G(\mathbf{r}, \mathbf{r}') ds' \right] ds \\
 &= \int_c \hat{\mathbf{n}} \cdot \left[\mathbf{f}_m(\mathbf{r}) \iint_{T_n} \nabla' \cdot \mathbf{f}_n(\mathbf{r}') G(\mathbf{r}, \mathbf{r}') ds' \right] dl \quad (2.52)
 \end{aligned}$$

By nature of the chosen basis function properties at contours, this goes to zero everywhere on the surface [8] leaving only the second expression of equation (2.51). This term can be substituted back into (2.48) which yields an equation where the del operator has been moved from the Green's function to the basis function,

$$\begin{aligned}
 & \sum_{n=1}^N I_n \left[\iint_{T_m} \mathbf{f}_m(\mathbf{r}) \cdot \iint_{T_n} \mathbf{f}_n(\mathbf{r}') G(\mathbf{r}, \mathbf{r}') ds' ds \right] \\
 & - \sum_{n=1}^N I_n \left[\frac{1}{k^2} \iint_{T_m} \left[\nabla \cdot \mathbf{f}_m(\mathbf{r}) \left(\iint_{T_n} \nabla' \cdot \mathbf{f}_n(\mathbf{r}') G(\mathbf{r}, \mathbf{r}') ds' \right) \right] ds \right] = \\
 & \quad \frac{-j}{\omega\mu} \iint_{T_m} \mathbf{f}_m(\mathbf{r}) \cdot \mathbf{E}_i(\mathbf{r}) ds \quad (2.53)
 \end{aligned}$$

this is the final representation of N linear equations with the del operator removed from the Green's function. The next section will finalise the EFIE MoM procedure by modifying the tested EFIE to be represented in matrix form.

2.3.4 RWG MoM formulation

Equation (2.53) is the discretized MoM solution of the EFIE. This section serves to stipulate the equation in matrix form after applying the RWG's special properties to the discretized equation. Let us stipulte equation (2.53) for the m th testing and n th

CHAPTER 2. FORMULATION OF THE 3D METHOD OF MOMENTS

source function as,

$$I_n \iint_{T_m} \iint_{T_n} \left[\mathbf{f}_m(\mathbf{r}) \cdot \mathbf{f}_n(\mathbf{r}') - \frac{1}{k^2} \nabla \cdot \mathbf{f}_m(\mathbf{r}) \nabla' \cdot \mathbf{f}_n(\mathbf{r}') \right] G(\mathbf{r}, \mathbf{r}') ds' ds$$

$$= \frac{-j}{\omega\mu} \iint_{T_m} \mathbf{f}_m(\mathbf{r}) \cdot \mathbf{E}_i(\mathbf{r}) ds \quad (2.54)$$

This is one equation in a set of N linear equations. The set of can be written as

$$[\mathbf{Z}] \{I\} = \{V\}$$

where an impedance matrix entry Z_{mn} can be written

$$Z_{mn} = \iint_{T_m} \iint_{T_n} \left[\mathbf{f}_m(\mathbf{r}) \cdot \mathbf{f}_n(\mathbf{r}') - \frac{1}{k^2} \right] G(\mathbf{r}, \mathbf{r}') ds' ds \quad (2.55)$$

and the excitation vector V_m is,

$$V_m = \frac{-j}{\omega\mu} \iint_{T_m} \mathbf{f}_m(\mathbf{r}) \cdot \mathbf{E}_i(\mathbf{r}) ds \quad (2.56)$$

at this point the equations are given for a basis function \mathbf{f}_m in general form. Recalling from section 2.3.2 the RWG basis function is specially designed for triangular mesh EFIE MoM solutions and equations (2.3.2) and (2.45) give these special properties. When these properties are applied to the impedance matrix and excitation vector it greatly simplifies the equation.

Figure 2.3 shows how an observer and source triangle interact. The Z_{mn} impedance matrix entry is dependant on four interactions between the respective triangles and can be written

$$Z_{mn} = \frac{l_m l_n}{A_m^+ A_n^+} \iint_{T_m^+} \iint_{T_n^+} \left[\frac{1}{4} \boldsymbol{\rho}_m^+(\mathbf{r}) \cdot \boldsymbol{\rho}_n^+(\mathbf{r}') - \frac{1}{k^2} \right] \frac{e^{-jkr}}{4\pi r} ds' ds$$

$$+ \frac{l_m l_n}{A_m^+ A_n^-} \iint_{T_m^+} \iint_{T_n^-} \left[\frac{1}{4} \boldsymbol{\rho}_m^+(\mathbf{r}) \cdot \boldsymbol{\rho}_n^-(\mathbf{r}') + \frac{1}{k^2} \right] \frac{e^{-jkr}}{4\pi r} ds' ds$$

$$+ \frac{l_m l_n}{A_m^- A_n^-} \iint_{T_m^-} \iint_{T_n^-} \left[\frac{1}{4} \boldsymbol{\rho}_m^-(\mathbf{r}) \cdot \boldsymbol{\rho}_n^-(\mathbf{r}') - \frac{1}{k^2} \right] \frac{e^{-jkr}}{4\pi r} ds' ds$$

$$+ \frac{l_m l_n}{A_m^- A_n^+} \iint_{T_m^-} \iint_{T_n^+} \left[\frac{1}{4} \boldsymbol{\rho}_m^-(\mathbf{r}) \cdot \boldsymbol{\rho}_n^+(\mathbf{r}') + \frac{1}{k^2} \right] \frac{e^{-jkr}}{4\pi r} ds' ds \quad (2.57)$$

and the excitation vector V_m can be written

$$V_m = \frac{-j l_m}{2\omega\mu A_m^+} \iint_{T_m^+} \boldsymbol{\rho}_m^+(\mathbf{r}) \cdot \mathbf{E}_i(\mathbf{r}) ds + \frac{-j l_m}{2\omega\mu A_m^-} \iint_{T_m^-} \boldsymbol{\rho}_m^-(\mathbf{r}) \cdot \mathbf{E}_i(\mathbf{r}) ds \quad (2.58)$$

CHAPTER 2. FORMULATION OF THE 3D METHOD OF MOMENTS

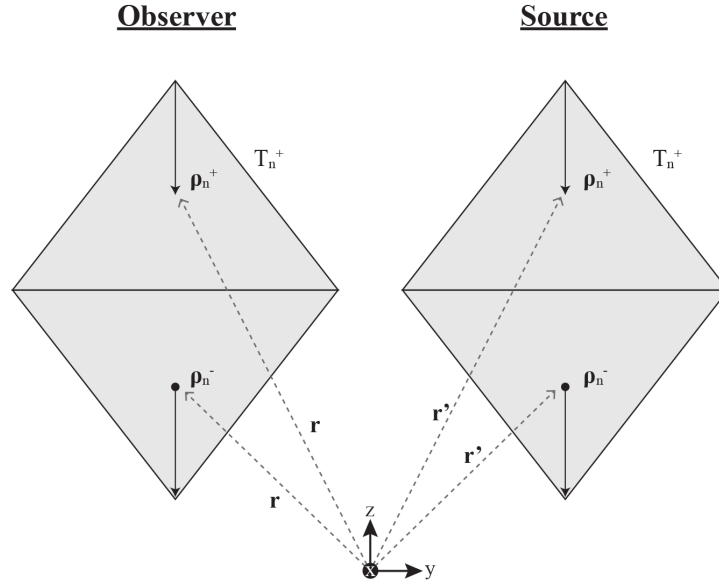


Figure 2.3: Interaction of an observer and source triangle

where r is the distance between an observer and source point on the triangle. These formulas illustrate how effective the RWG MoM method can be, as each impedance matrix entry only requires four complex multiplications.

A large number of integrations must be carried out over the solution domain. This can be performed by the use of Gaussian quadrature for triangular domains. This piece specifically makes use of the quadrature methods presented in [12]. When the observer T_m^\pm and source triangles T_n^\pm are very close to one another the Green's function of the integrand approaches a singularity. This can either be addressed by extracting and solving the singularity analytically such as presented in [13] or by applying a singularity cancellation transformation method such as that presented in [14]. The method presented in [14] is employed in the development of this MoM solver.

2.3.5 Conclusion

Although the process of moving from basic Maxwell's equations to the EFIE MoM can be found in literature, its presentation can be somewhat disjointed and its development unclear. This work's goal is to serve as a basis for future work. Therefore this chapter sought to stipulate the aforementioned process in detail, highlighting and emphasising

CHAPTER 2. FORMULATION OF THE 3D METHOD OF MOMENTS

the steps that are often presented somewhat vaguely.

The presented works application to a highly accurate MoM solver is now be presented in detail.

Chapter 3

Development of a MATLAB based MoM library

The aim of this project described in chapter 1 was the development of an efficient and highly accurate MoM solver to serve as a basis for future work in the Stellenbosch CEM research group. At the point of writing the following research papers and articles that made use of the code as a basis have been published or have been submitted for publishing

- *K. Serwaj, M. M. Botha "Computation of MBF Reaction Matrices for Antenna Array Analysis, with a Directional Method", [15]*
- *M. Chose, M.M. Botha "Improvements to the Domain Green's Function Method for Antenna Array Analysis", [16]*
- *M. Chose, M.M. Botha "An iterative method for the analysis of large disjointed antenna arrays", submitted for EUCAP 2020*
- *K. Serwaj, M. M. Botha "Directional Method to Compute Reduced Matrix System in MBF Solvers", submitted for EUCAP 2020*

The solver is currently in use by two PHD students, one masters student and has been modified by another for special use. Three final year students have employed the code in completing their undergraduate degrees. The solver has already proven its usefulness in forming a platform on which researchers can build specialised methods (using its many features) to test basis function methods, error estimation, ACA and a multitude of other CEM techniques.

This chapter is a highly detailed description of the solver and its functionalities. The idea of this chapter is to serve as a developers manual for future expansions and customisations of the solver. An emphasis must be placed on the fact that this is not a basic implementation of a simple MoM code, but rather a solver built with the

CHAPTER 3. DEVELOPMENT OF A MATLAB BASED MOM LIBRARY

capability of solving mesh structures with extremely high accuracy. Much thought and effort went into fully optimising the solver while keeping it single threaded and relatively easy to understand.

This chapter is split into four primary sections. The first section details the solver system by following the procedure of solving for a simple mesh and explaining the inner workings of the system as the given mesh progresses through the various stages of the solver. Secondly, the actual process of solving for the impedance matrix within the MEX is described in detail. Lastly, the accuracy of MoM engine is compared FEKO.

3.1 The MoM Solver

3.1.1 Raw mesh data description

The solver has a built in function called *importNastran.m* for processing a NASTRAN file, which is a convenient way of importing mesh data. Many EM programs such as FEKO and Gmsh can export data in .nas format which gives the benefit of being able to setup complex meshes in their user friendly graphical user interfaces and solve them locally. The solver does not require the data to be extracted from the NASTRAN format, any meshing tools will suffice, as long as a list of triangles and coordinates are provided the solver can proceed. If a user wanted to implement and test a new feature the user can create an example mesh of only a few triangles by hand, or using a custom script. Figure 3.1 shows the how *importNastran.m* will process and return the data of a single RWG triangular pair.

This set up is often referred to as a finite connectivity set [1]. It is made up of a node list and a set of coordinates. Each row in the node list is a set of three nodes that refer to row indices of the coordinate set. Finite connectivity lists can be used to represent extremely complicated structures using only triangular mesh elements as triangles can conform to most complicated surfaces.

Once a finite connectivity list is defined RWG basis functions such as those described in section 2.3.2 will be defined over the mesh using the *EdgeCalc.m* routine.

3.1.2 Practical implementation of RWG basis functions

The *EdgeCalc.m* routine defines and numbers RWG basis functions over a given connectivity set using a custom data structure convention. The resultant data structure

CHAPTER 3. DEVELOPMENT OF A MATLAB BASED MOM LIBRARY

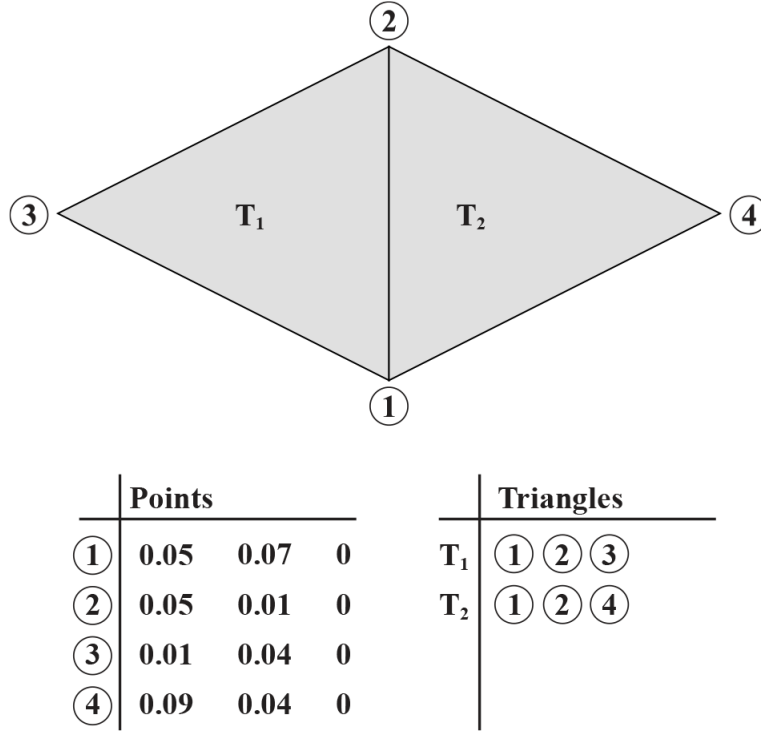


Figure 3.1: Data outputted by *importNastran.m* or required for custom mesh setup

is eventually sent to the MoM solver. This section defines the custom data structure that will be returned by *EdgeCalc.m*. The succeeding section will describe how *EdgeCalc.m* goes about setting up the data structure.

When deciding on how RWG's should be defined using the connectivity set a many factors must be taken into consideration which include,

- Basis functions can only be defined over internal edges of the mesh structure.
- Current directions must remain consistent throughout the mesh system, when current is defined as travelling out of a triangle the triangle sharing that edge must accommodate it by defining current as travelling inwards.
- Due to the MoM's computational complexity (N^2) the process of labelling the mesh's RWG basis functions must have a computational complexity as close N as possible, minimizing runtime overhead.

CHAPTER 3. DEVELOPMENT OF A MATLAB BASED MOM LIBRARY

- The basis functions must be uniquely and consistently numbered as this will determine its Z -matrix position.
- It must be as simple as possible to allow for easy understanding and modification.

With these factors in mind a convention for the data structure describing a RWG's geometric properties is defined as,

$$A \quad B \quad C \quad || \quad D \quad E \quad F \quad || \quad G \quad H \quad I$$

where A, B, C are the original triangle nodes with six new values appended to them. D, E, F and G, H, I define properties relating to the nodes. F, E and D are either 1 or -1 determining current direction over edges AB , AC and BC respectively. Where positive 1 means current flows away from the free vertex and -1 towards it. Entries I, H and G are either a positive integer defining the basis function number over the corresponding edge or -1 meaning there is no basis function defined on the corresponding edge.

To illustrate this data structure setup, basis functions are defined over the mesh given in figure 3.1. Triangle \mathbf{T}_1 is retained and displayed in figure 3.2 along with an algebraic interpretation of the same triangle.

The nine numbers below the triangles in figure 3.2 is the data of a single triangle. Only one of the edges on this triangle has a basis function associated with it. The basis function exists over edge $\textcircled{1}$ $\textcircled{3}$ and is assigned the number 1. For the rest of this piece basis function numbers will be wrapped in a square as in figure 3.2 unless otherwise stated.

A data structure for defining the geometric properties of the RWG basis functions is defined with only six additional values appended to each triangle. As the geometric definition of the RWG basis functions are now defined, the way in which the *EdgeCalc.m* routine goes about assigning them will be reviewed.

3.1.3 Edge finding algorithm

The functioning of the *EdgeCalc.m* routine is now described via a simple example. *EdgeCalc.m* processes the triangles and points data to set up the data structure defined in the previous section.

CHAPTER 3. DEVELOPMENT OF A MATLAB BASED MOM LIBRARY

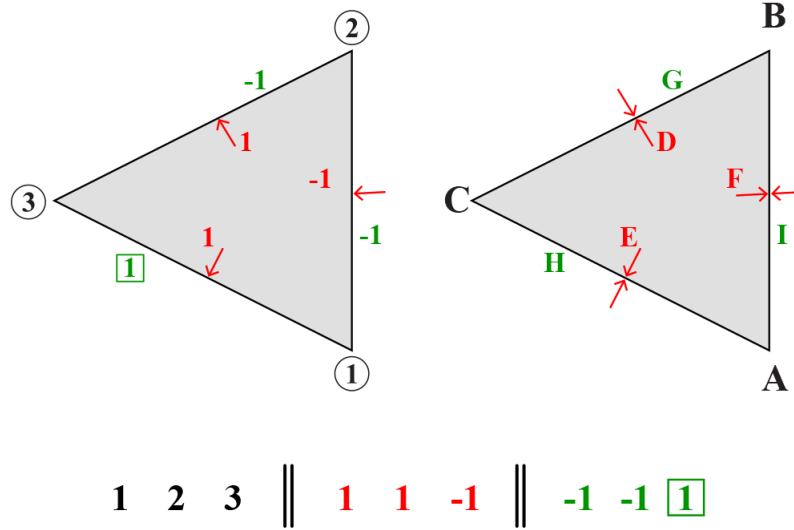


Figure 3.2: Illustration of the MoM solver's data structure convention

Figure 3.3 illustrates a small plate of four triangles along with its connectivity list. *EdgeCalc.m*'s first function is setting up this connectivity list. Each column in the connectivity list is a group of triangles that are connected to the node designated by a circled number at the top of the list. It is a common method used in mesh processing as it avoids the requirement to store and search large data sets.

The *EdgeCalc.m* method then proceeds to simultaneously number the basis function existing over each inner edge and define the direction of positive current flow. Triangle \mathbf{T}_1 is initially defined as

$$\begin{array}{ccccccc} \textcircled{1} & \textcircled{2} & \textcircled{3} & || & 0 & 0 & 0 & || & 0 & 0 & 0 \end{array}$$

The algorithm then chooses each edge in the triangle, $\textcircled{1} \textcircled{2}$, $\textcircled{2} \textcircled{3}$ and $\textcircled{1} \textcircled{3}$ individually. It uses the edge nodes as column entry points to the connectivity list and

CHAPTER 3. DEVELOPMENT OF A MATLAB BASED MOM LIBRARY

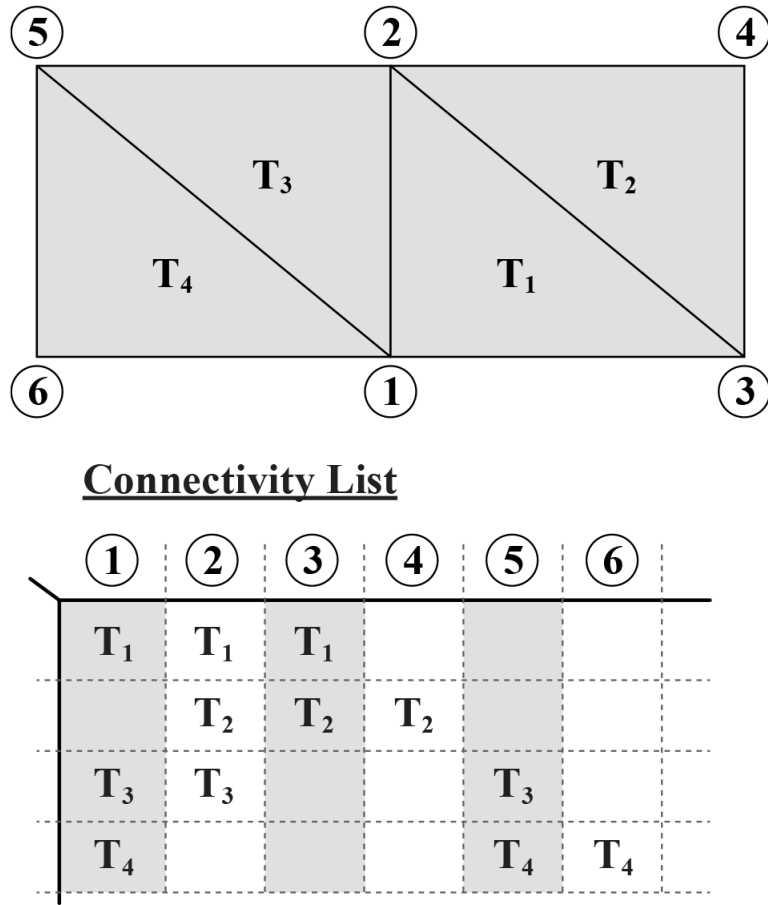


Figure 3.3: Small PEC plate with connectivity list

finds the intersection of triangles defined in the lists. Choosing edge ① ②

①	②
T_1	T_1
T_3	T_2
T_4	T_3

These two columns intersection is T_1 and T_3 . The edge is shared by two triangles therefore a basis function must exist over it and must be assigned a number and direction. This is achieved by again observing the intersection. The data is set up in a way

CHAPTER 3. DEVELOPMENT OF A MATLAB BASED MOM LIBRARY

that higher indexed triangle always appears second in the intersection. If the selected triangle, \mathbf{T}_1 , is the lower indexed triangle it means this is the first time edge ① ② is encountered. The current direction over the edge is assigned 1 for outward flowing and the basis function is given a new number, 1 in this case. If the selected triangle is the higher indexed of the two, the current direction is assigned -1 and *EdgeCalc.m* must retrieve the basis function number from the lower indexed triangle where it has already been assigned a value. \mathbf{T}_1 will now appear as

$$\textcircled{1} \quad \textcircled{2} \quad \textcircled{3} \quad || \quad 0 \quad 0 \quad 1 \quad || \quad 0 \quad 0 \quad \boxed{1}$$

The process is repeated for the remaining triangles until the list is complete. *EdgeCalc.m* returns the new triangle data and the basis supports. The basis supports are the two triangles over which a basis function is defined. When an intersection has two values such as \mathbf{T}_1 and \mathbf{T}_3 it is appended to a list of basis supports. The final basis support list is

Basis supports			
1	→	\mathbf{T}_1	\mathbf{T}_3
2	→	\mathbf{T}_1	\mathbf{T}_2
3	→	\mathbf{T}_3	\mathbf{T}_4

Figure 3.4 shows the final designation of basis function numbers and current direction.

The *EdgeCalc.m* algorithm has a computational complexity very close to N and sets up the data required for the MoM solution.

3.1.4 Common basis function selection and mesh reduction

A primary goal in developing this MoM solver was the ability to select clusters of source and observer basis functions and solve for them specifically, avoiding ever having to solve for large Z -matrices. It is also not ideal for every student to write a MoM code with this functionality as developing a solver to perform this task efficiently (without ever searching arrays) is not trivial. This section describes the processing performed by *BasisFunctionSelect.m* in preparing and editing the mesh data to allow for basis function selection.

The *main.m* file of the MoM solver has two input values, *ObsBasisSelect* and *SrcBasisSelect*. These inputs are arrays in which the user can specify the source and basis functions the user would like to be solved for independently.

CHAPTER 3. DEVELOPMENT OF A MATLAB BASED MOM LIBRARY

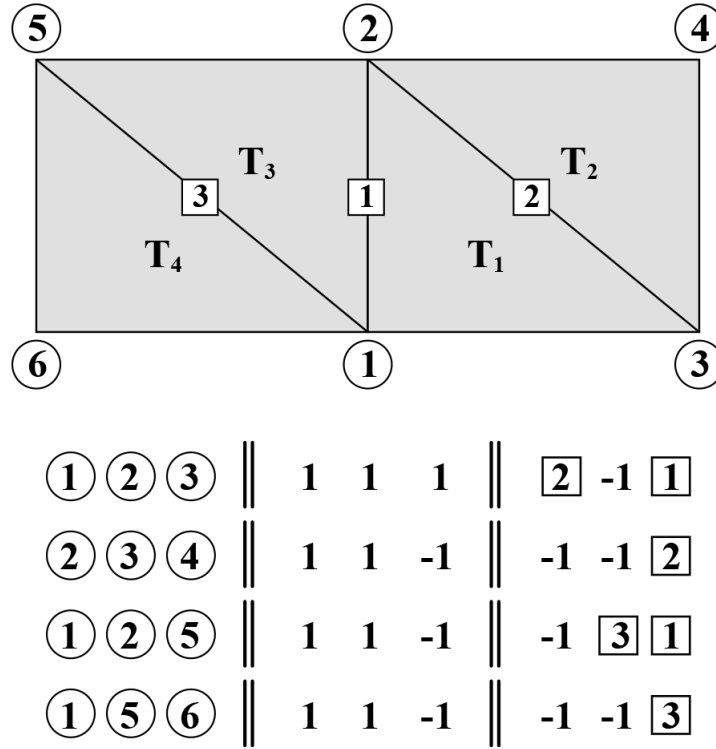


Figure 3.4: Small PEC plate with connectivity list

BasisFunctionSelect.m receives the selected source and observer lists and takes their union, it then creates a new mesh with this data as a first optimization. Taking figure 3.4 as the starting mesh basis function $\boxed{1}$ is selected as a source and observer and $\boxed{3}$ is chosen as an observer only. It can be represented as,

Basis function index	obs flag	src flag
$\boxed{1}$	1	1
$\boxed{2}$	0	0
$\boxed{3}$	1	0

where a 1 in the column means it is selected as observer or source. The union of this selection is obviously $\boxed{1}, \boxed{3}$. *BasisFunctionSelect.m* uses the union numbers as entry points to the previously defined basis support list.

CHAPTER 3. DEVELOPMENT OF A MATLAB BASED MOM LIBRARY

Basis supports					
•	1	→	T₁	T₃	
	2	→	T₁	T₂	
•	3	→	T₃	T₄	

From this it determines which triangles have basis functions defined over them and creates a triangle list, discarding the triangles over which basis functions do not exist. The new triangle list now appears as

$$\begin{array}{ccc}
 \textcircled{1} & \textcircled{2} & \textcircled{3} \\
 \textcircled{1} & \textcircled{2} & \textcircled{5} \\
 \textcircled{1} & \textcircled{5} & \textcircled{6}
 \end{array}
 \parallel
 \begin{array}{ccc}
 1 & 1 & 1 \\
 1 & 1 & -1 \\
 1 & 1 & -1
 \end{array}
 \parallel
 \begin{array}{ccc}
 \boxed{2} & -1 & \boxed{1} \\
 -1 & \boxed{3} & \boxed{1} \\
 -1 & -1 & \boxed{3}
 \end{array}$$

although the triangle list no longer contains unused triangles the nodes in the triangle list still refer to a full points list and the basis functions are incorrectly numbered. The nodes and basis functions must be renumbered to create a completely independent mesh. In order to avoid ever performing a search of N^2 complexity a node and basis function map are created. The node map is a $4N$ complexity to create and the basis function map can be taken from the aforementioned basis support union,

Node Map			Basis function map		
$\textcircled{1}$	→	$\textcircled{1}$	$\boxed{1}$	→	$\boxed{1}$
$\textcircled{2}$	→	$\textcircled{2}$	$\boxed{2}$	→	0
$\textcircled{3}$	→	$\textcircled{3}$	$\boxed{3}$	→	$\boxed{2}$
$\textcircled{4}$	→	0			
$\textcircled{5}$	→	$\textcircled{4}$			
$\textcircled{6}$	→	$\textcircled{5}$			

The two maps indicate the node and basis function numbering of the new mesh. *BasisFunctionSelect.m* uses these two maps to renumber their respective sources. A value in the triangle list gives the index into the respective maps, thus, avoiding any N^2 searches.

Figure 3.5 shows the resulting completely independent mesh, which can be operated on as if it is a new problem. With this optimisation the Z -matrix mesh does not have to iterate over the entire mesh, searching for which basis functions are active.

At this point a new mesh is created and would solve for a 2×2 impedance matrix. It does not take into account that basis function $\boxed{2}$ (originally $\boxed{3}$) is only an observer. The amount of computations can be halved if this is implemented efficiently. This is achieved by creating two lists *SrcMap* and *ObsMap*. These lists are sent with the triangles and point data to the *mom.c* solver. They are length N and appear as,

CHAPTER 3. DEVELOPMENT OF A MATLAB BASED MOM LIBRARY

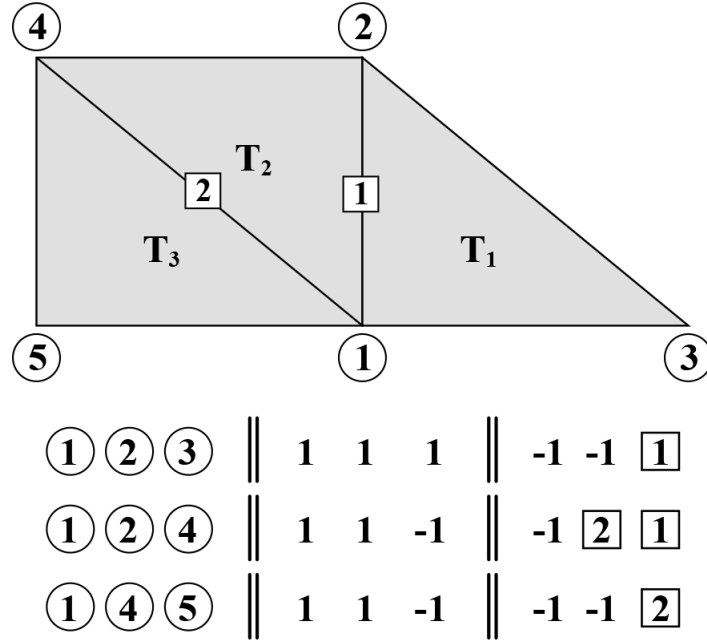


Figure 3.5: New mesh created by *BasisFunctionSelect.m*

Observer map		Source map	
$\boxed{1}$	\rightarrow	$\diamond 1$	
$\boxed{2}$	\rightarrow	$\diamond 2$	
		$\boxed{1}$	\rightarrow
		$\boxed{2}$	\rightarrow
			0

The number enclosed in the diamond indicate what the corresponding basis functions index will be in the resultant $M \times N$ matrix. If an entry of either *SrcMap* or *ObsMap* is zero *mom.c* will skip the calculation. *mom.c* interprets the basis function \boxed{x} as an index to *SrcMap* or *ObsMap* which will indicate the Z_{mn} position in the $M \times N$ matrix. This is illustrated in figure 3.6,

At this point *BasisFunctionSelect.m* has fully prepared the data to be provided to *mom.c*. The *mom.c* routine makes up MoM solver's core and will subsequently be introduced and described in detail.

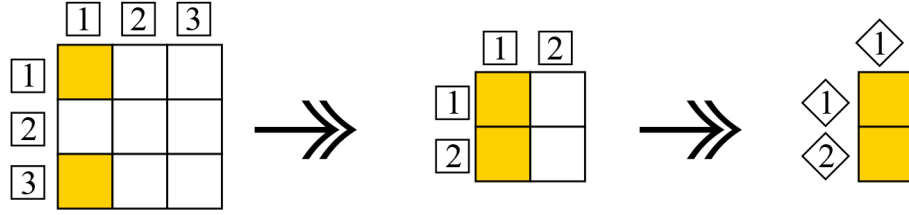


Figure 3.6: Process of basis function renumbering

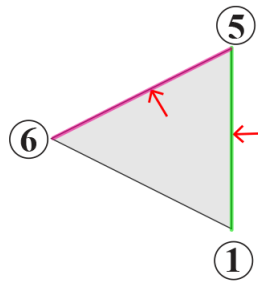
3.2 Impedance matrix solution

This section covers the fundamental steps employed by the **MeX** function *mom.c* to accurately solve for the Z -matrix entries while operating with a reasonable run time. As a fundamental goal of this solver was its ease of use and understandability the data structures are kept as simple as possible. Higher level coding paradigms such as piping, hash tables and classes are avoided.

3.2.1 Solver design

This section discusses important steps and logic in the code. Figure 3.7 illustrates the ten main levels and logic steps in the code block. The sections numbered in black indicate standard MoM functionality and sections numbered in red are custom checks for optimisation and basis function selection. An explanation of these steps follows.

- 1 Chooses an observer triangle $pTri$. Coloured lines on the triangle indicate edges over which basis functions are defined.



$$\textcircled{1} \quad \textcircled{5} \quad \textcircled{6} \parallel 1 \quad 1 \quad -1 \parallel \boxed{4} \quad -1 \quad \boxed{3}$$

CHAPTER 3. DEVELOPMENT OF A MATLAB BASED MOM LIBRARY

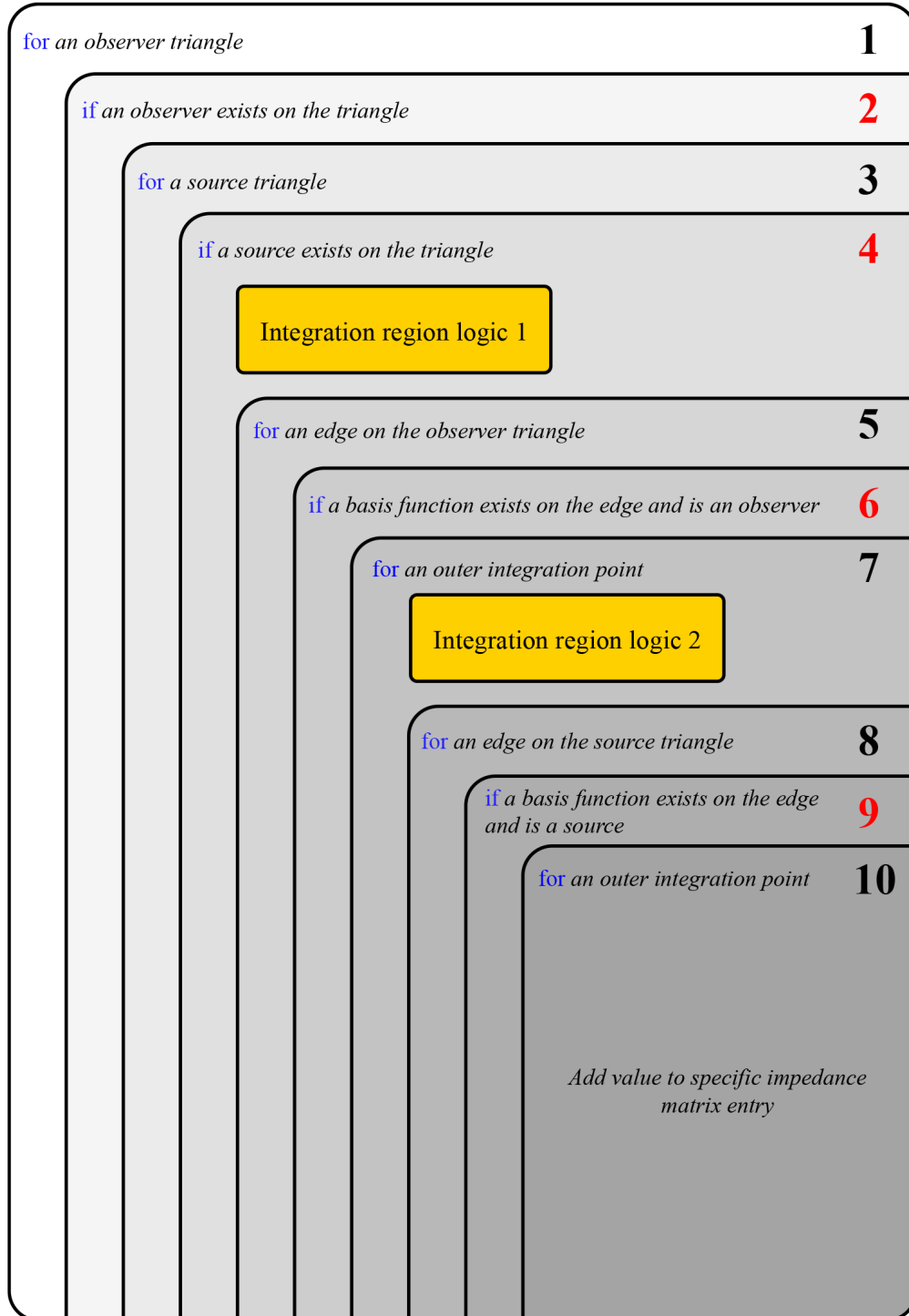


Figure 3.7: MoM solver code representation

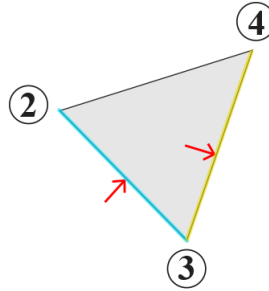
CHAPTER 3. DEVELOPMENT OF A MATLAB BASED MOM LIBRARY

- 2** A custom step required purely for optimization purposes. When a technique such as ACA is used it extracts entire rows or columns from the impedance matrix. It will, for example, have N sources and 1 observer, this means optimization by mesh reduction covered in 3.1.4 does not have an effect due to the entire mesh being retained. Observing the following code in the **MeX**

```
1270 int pbasisindex[3] = {pTri[6], pTri[7], pTri[8]};
1271 if (ObserverOnTriangle(pbasisindex, obs_map))
1272
```

it extracts the basis function data $\left(\begin{bmatrix} 4 \end{bmatrix} \parallel -1 \parallel \begin{bmatrix} 3 \end{bmatrix} \right)$ and checks if one of the basis functions are defined as an observer in the *ObsMap* array. It stops the step **3** from looping over testing triangles as no basis function is defined over the selected observer.

- 3** Choose a source triangle $qTri$. Coloured lines on the triangle indicate edges over which basis functions are defined.



$$\begin{pmatrix} 2 & 3 & 4 \end{pmatrix} \parallel \begin{pmatrix} 1 & 1 & -1 \end{pmatrix} \parallel \begin{pmatrix} 1 & -1 & 2 \end{pmatrix}$$

- 4** This step blocks loops that calculate nothing from occurring. It is identical to step **2** except it uses the source triangles basis functions $\left(\begin{bmatrix} 1 \end{bmatrix} \parallel -1 \parallel \begin{bmatrix} 2 \end{bmatrix} \right)$ as entry points to *SrcMap* checking if the basis functions are sources. The code is as follows,

```
1284 int qbasisindex[3] = {qTri[6], qTri[7], qTri[8]};
1285 if (SourceOnTriangle(qbasisindex, src_map))
1286
```

CHAPTER 3. DEVELOPMENT OF A MATLAB BASED MOM LIBRARY

An issue with earlier versions of the code was that calculating a single impedance matrix column would take between 10 and 20 times longer than calculation of a row. This would occur as MoM code is a row implementation, it picks an observer triangle and cycles over all the triangles checking for source basis functions. It then picks the next observer triangle and repeats the process. If a single Z -matrix column is a desired solution the code would select an observer and search all the triangles for possible source basis functions and only ever calculate a single entry. Approximately $(N - 1)^2$ cycles are performed where nothing is calculated, this optimization reduces that amount to 3 as it simply checks if a source exists on the triangle before anything else happens. The gold block that can be observed in 3.7 represents the stage at which the outer integration points are chosen. At this point it is known that at least one integration must be performed. Section 3.2.2 will expand on this further.

- 5 This part simply chooses an edge of the observation triangle.

```
1554   for ( ii=0; ii < 3; ii++) //Edges of observation triangle
1555
```

an arbitrary observer triangles edge is chosen as



- 6 As an edge is selected the solver must determine two things, is there a basis function defined over the edge and is it an observer. The code

```
1569   int pEdgeIndex = pTri[6+( ii+2)%3];
1570   int p_obs_index = obs_map[ ( pTri[6+( ii+2)%3]-1) ];
1571   // this if statement asks two questions, is it a basis function?
      and is it an observer?
```

CHAPTER 3. DEVELOPMENT OF A MATLAB BASED MOM LIBRARY

```
1572  if ((pEdgeIndex + 1) && (p_obs_index + 1))
1573
```

finds the relevant edge's basis function number $\boxed{5}$ and uses it as an entry point to the *ObsMap* array,

Observer map		
$\boxed{1}$	→	$\diamond 1$
$\boxed{2}$	→	$\diamond 2$
$\boxed{3}$	→	$\diamond 3$
$\boxed{4}$	→	0
• $\boxed{5}$	→	$\diamond 4$

if both values are positive integers as in this case $\boxed{5}$ and $\diamond 4$ it proceeds knowing a basis function exists on the edge and it is also an observer.

- 7 A standard MoM solver function. Recalling from section 2.3.4 the integration is performed using Gaussian quadrature for triangular domains, the double integration results in a nested sum. The source quadrature must be performed for every quadrature point on the (observer) outer triangle.

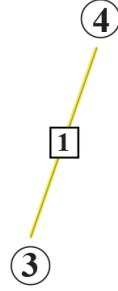
```
1569  for (oip=0; oip<NumOuterIntPoints; oip++) //outer integral points
1570
```

- 8 With the outer integration points defined an edge of the source triangle is chosen over which a basis function could possibly exist.

```
1554  for (jj=0; jj<3; jj++) //Edges of source triangle
1555
```

A source triangles edge is chosen as

CHAPTER 3. DEVELOPMENT OF A MATLAB BASED MOM LIBRARY



- 9 With the edge selected on the inner triangle it must be determined if the edge has a basis function defined over it and if it is a source. The code

```

1569  int qEdgeIndex = qTri[6+(jj+2)%3];
1570  int q_src_index = src_map[(qTri[6+(jj+2)%3]-1)];
1571  if ((qEdgeIndex + 1) && (q_src_index + 1))
1572

```

finds the relevant edge's basis function number $\boxed{1}$ and uses it as an entry point to the *SrcMap* array,

Source map			
•	$\boxed{1}$	→	$\diamond 1$
	$\boxed{2}$	→	$\diamond 2$
	$\boxed{3}$	→	$\diamond 3$
	$\boxed{4}$	→	0
	$\boxed{5}$	→	$\diamond 4$

if both values are positive integers as in this case $\boxed{1}$ and $\diamond 1$ it proceeds knowing a basis function exists on the edge and it is also a source.

- 10 The final level is the source triangles integration using Gaussian quadrature. For each point chosen in the code as

```

1713  for (iip=0; iip < newNumInt; iip++)
1714

```

CHAPTER 3. DEVELOPMENT OF A MATLAB BASED MOM LIBRARY

the weighted effect is summed to the relevant entry of the $M \times N$ impedance matrix. For the presented example source basis function $\boxed{1}$'s effect on observer $\boxed{5}$ is placed in position $Z \left[\diamond 4, \diamond 1 \right]$ of the $M \times N$ custom matrix.

This section broke down the MoM solver's logic in performing the integrations required for the impedance solution. The implementation's custom logic ensures efficiency by never calculating values or assigning memory when it is not required. The following section explains the process of dynamically assigning integration points per calculation.

3.2.2 Integration regions

As discussed in section 2.3.4 Gaussian quadrature is used to perform the triangular integration when the source and observer triangles are far apart. The scheme chosen to handle singular and near singular matrix values was the *Radial – Angular – R^1 – Sqrt* (from now on referred to as RAR) method [14], a singularity cancellation transformation rather than an analytical extraction.

Two separate functions were developed to handle the standard Gaussian quadrature and singularity cancellation quadrature respectively. They appear in the code as follows, Gaussian Quadrature

```
416 void GaussianQuadrature(double points[][3], int numIntPoints, double **
    returnPoints)
```

and RAR as,

```
779 void RARIS(double points[][3], double obsPoint[], int numIntPoints, double
    **returnPoints)
```

although implementation of these functions was far from trivial they are not the focus of this piece and thus their implementation will not be thoroughly discussed. Having access to these two functions alone would save a researcher much effort.

The only difference in what they receive is the observation point as *RARIS* requires the point to perform the singularity cancellation. The common inputs to these functions is *numIntPoints* and *returnPoints*. Observing figure 3.7 integration region logic 1 and 2 are defined. At logic 1 the amount of outer integration points are assigned memory and chosen, at logic 2 the amount of inner integration points are assigned memory and

CHAPTER 3. DEVELOPMENT OF A MATLAB BASED MOM LIBRARY

chosen. In optimizing a MoM solver assigning integration points as late as possible is greatly beneficial as it allows for the checks discussed in the preceding section to stop as many unnecessary loops as possible.

Assigning the correct amount of points for a quadrature integration is crucial in attaining an extremely accurate MoM solver, but assigning too many integration points yields an unnecessarily high computational overhead. This MoM solver goes to great lengths in ensuring enough integration points are assigned to accurately compute the Z -matrix while ensuring the solution is not overly expensive computationally.

When *mom.c* reaches logic 1 in figure 3.7 it has selected an observer and source triangle, it is certain that at least a single integration will occur. Outer integration points must be allocated at this point and the number of inner integration points is either selected or a flag is set that indicates singularity cancellation is possibly required.

The MoM solver uses the maximum diameter of the observer and source triangles as a metric against which to test the distance between them for integration point selection. Observing figure 3.8 the decimal numbers indicate the distance between the source and observers ratio to their common maximum diameter.

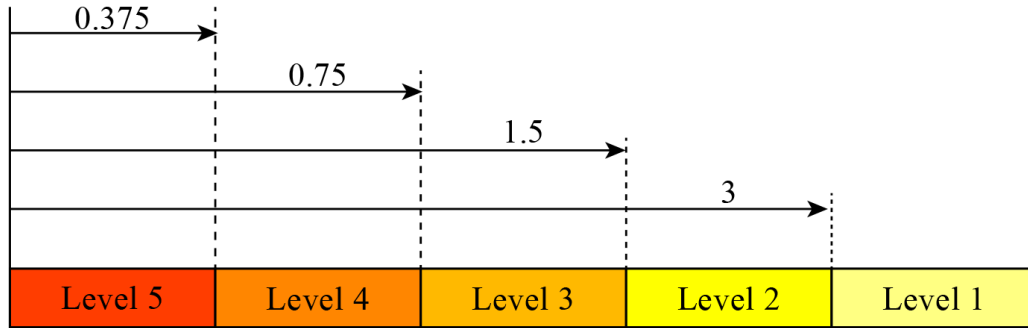


Figure 3.8: Levels of integration accuracy

The maximum amount of Gaussian quadrature available in the engine is 33, the RAR the radial multiplied by the angular cannot exceed 300 points. Both are higher than ever realistically required. The MoM solver has four modes of operation each with increasing solution accuracy.

Table 3.1 shows the amount of integration points assigned to the triangles for the various levels. The two numbers per entry denote the amount of inner and outer in-

CHAPTER 3. DEVELOPMENT OF A MATLAB BASED MOM LIBRARY

	Level 5	Level 4	Level 3	Level 2	Level 1
MODE 0	7 <i>RAR</i>	6 6	4 4	3 3	3 3
MODE 1	13 <i>RAR</i>	7 7	6 6	4 4	3 3
MODE 2	16 <i>RAR</i>	12 12	7 7	6 6	3 3
MODE 3	25 <i>RAR</i>	16 16	12 12	7 7	3 3

Table 3.1: Selectable integration modes

tegration points per triangle respectively. The inner and outer amount of integration points are the same on all levels except for level 5 where *RAR* is used on the inner triangle, this is specifically to allow for a highly symmetrical impedance matrix. Symmetry in the Z -matrix is crucial when testing optimization methods like ACA a matrix partitioning scheme that takes advantage of weak coupling between well separated blocks of the impedance matrix, these are blocks that exist far away from the matrix diagonal. The impedance matrix is by definition symmetrical over its diagonal, but the use of Gaussian quadrature can cause symmetrical elements to differ slightly when the inner and outer integrals are changed for the observer and source basis function swapping roles.

The code flow for integration region logic 1 is displayed in figure 3.9. Three distance calculations are used in the solver; $d_{conservative}$ which estimates distance between the triangles loosely as,

$$d_{conservative} = (2/3) \times (D_1 + D_2), \quad (3.1)$$

where D_1 D_2 are the triangles respective diameters. d_9 finds the shortest distance between the triangle vertices and d_{36} which finds the shortest distance between the vertices and edge midpoints. In order to avoid using unnecessarily complicated checks the higher accuracy distance calculations are nested in the lower. Thus, higher accuracy checks are not done for triangles that are far apart. The only time when the amount of inner integration points is not assigned in logic 1 is when the source and observer triangles are either touching or they are very close together, in which case inner integration points will be allocated per outer integration point to ensure any singularity is dealt with accordingly. If the amount of inner integration points are not assigned this code block sets the *further check required* flag to 1. Figure 3.9 is the code flow diagram for integration region logic 2 from figure 3.7. The *proximity* variable is a highly

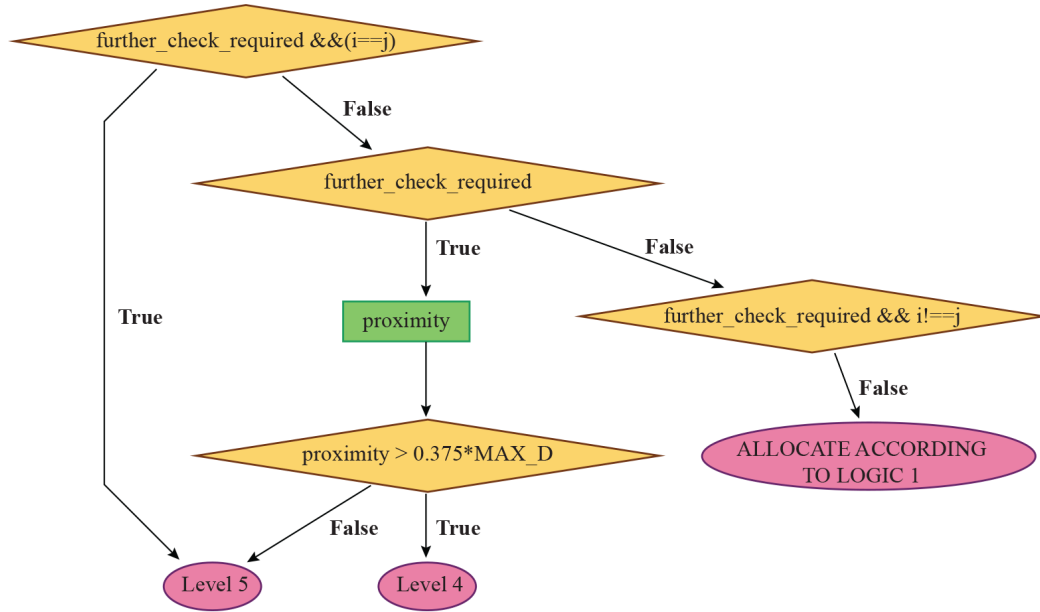


Figure 3.10: Integraion logic 2

accurate approximation of the outer integration point distance to the source triangle, it determines whether singularity cancellation or simply a high amount of integration points is required. After logic 2 is complete the amount of points and memory of the inner and outer integrals is allocated.

This section detailed the intricacies of selecting the correct number of integration points for specific integration regions. The subsequent sections benchmark tests the solver against FEKO.

3.3 MoM solver results and benchmark testing

At this point the solvers Core functions have been implemented. This section tests their capabilities using FEKO as a reference. The three main factors tested are the solvers convergence as the integration accuracy is increased, the solvers ability to handle problems with strong near singularities and the solvers runtime.

CHAPTER 3. DEVELOPMENT OF A MATLAB BASED MOM LIBRARY

custom 1	custom 2	custom 3	custom 4	custom 5	custom 6	custom 7	custom 8
3 RAR	4 RAR	6 RAR	7 RAR	12 RAR	13 RAR	16 RAR	25 RAR
3	3	4	6	7	12	13	16
3	3	3	4	6	7	12	13
3	3	3	3	4	6	7	12
3	3	3	3	3	4	6	7
108.7038	107.2561	102.7492	101.8303	101.1457	101.0674	100.4336	100.5754

Table 3.2: Integration points used to test convergence

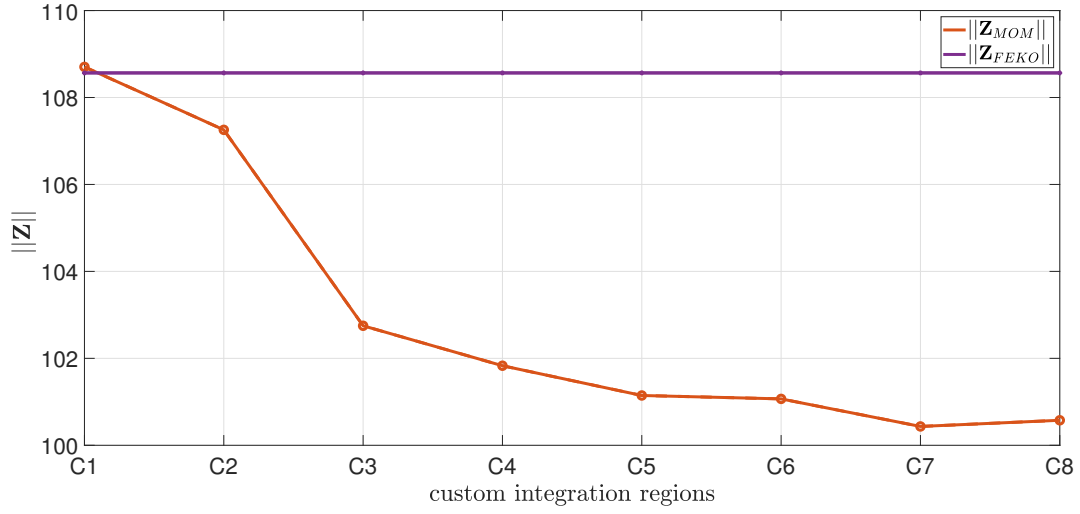
3.3.1 Convergence of solution

In order to test the solver's convergence a custom set of integration point allocations was written into the solver that steadily increased the amount of integration points to a maximum. The points are presented in table 3.2. The test is applied to a sphere of 1094 triangles. The metric used to test the convergence is the frobenius norm. Figure 3.11 plots the Frobenius of each matrix against the amount of integration points. When the MoM solver's integration points are set to the lowest possible value it correlates the most with the MoM solution, as the integration accuracy is increased the MoM solvers matrix norm convergers to a value other than FEKO. The values are given at the bottom of each column in table 3.2. FEKO's value is $\mathbf{Z}_{FEKO} = 108.5654$ almost identical to the lowest solver setting. The MoM solver integrates more accurately than FEKO, this not surprising as FEKO is an industrial tool and must have an extremely low runtime. Thus, when working with highly sensitive data where computational speed is not an issue it would be beneficial to work with this MoM solver.

3.3.2 Near singularity problems

This section tests the MoM solvers ability to handle singularities. This is tested by placing two parallel PEC plates in the presence of a plain wave and moving them closer together. It is known from the previous section that when the MoM solver is set to an extremely high accuracy as with these plates, the MoM solver and FEKO norms are different, but by a constant factor and if the plates move apart the factor should stay constant. Figure 3.12 illustrates how the norms change when the plates come closer together, but they stay relatively constant, when the plates are very close together.

CHAPTER 3. DEVELOPMENT OF A MATLAB BASED MOM LIBRARY

Figure 3.11: \mathbf{Z}_{MoM} convergence to a norm value

	0.0001λ	0.001λ	0.01λ	0.1λ
Entire matrix	-8.0367	-5.6868	-5.7667	-5.7753
Central block	-7.7667	-5.7579	-5.7593	-5.7583
Off diagonal block	-5.4359	-5.4360	-5.4351	-5.4366

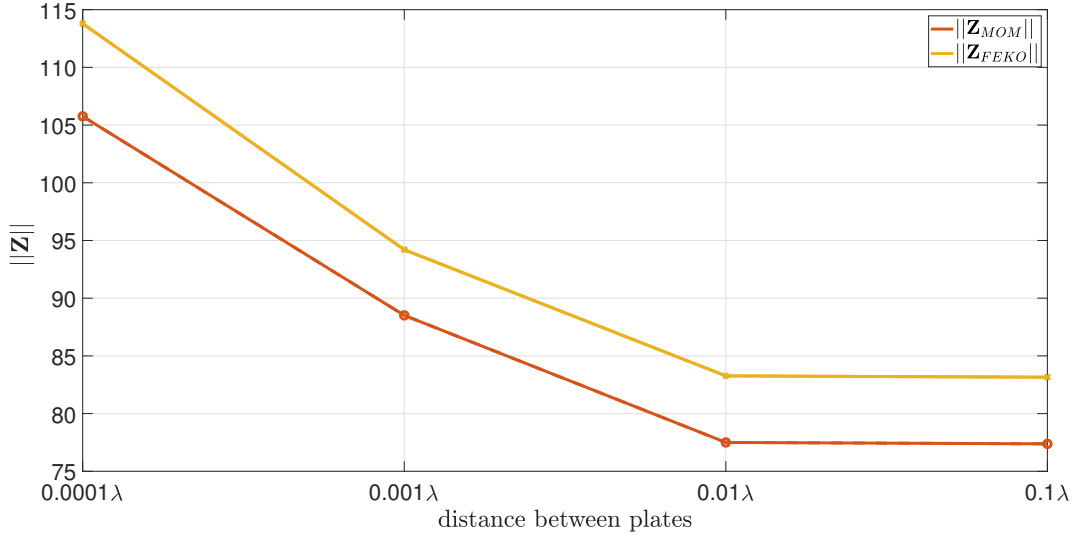
Table 3.3: Difference between FEKO and MoM solver matrix norms at near singularity points

Table 3.3 shows the error between FEKO and the MoM solver increases somewhat but stay very close together. This proves that even though there is a difference in the singularity cancellation method employed by the two solvers the MoM solver handles the singularity very well.

3.3.3 Runtime Analysis

The previous two sections illustrated the accuracy of the MoM solver is sufficient for research. Although accuracy is the primary concern for a MoM solver it must be able to perform simulations within reasonable run times. This section presents an analysis of the MoM solver run times. All of the runtimes referred to in this code are specific to the matrix fill. The connectivity data preprocessing is negligible in runtime compared to the matrix fill.

CHAPTER 3. DEVELOPMENT OF A MATLAB BASED MOM LIBRARY

Figure 3.12: Z_{MoM} norm as parallel plates are moved closer to each other

Amount of unknowns	FEKO (seconds)	MoM code (seconds)
684	0.523	4.788
855	0.796	6.5417
1026	1.051	8.4794
1196	1.32	10.8618
1368	1.584	13.4247

Table 3.4: MoM solver runtime for increasing amounts of unknowns

As FEKO is an industrial grade MoM solver it will be much more efficient than the custom code. The implemented MoM code is still relatively fast, table 3.4 shows a comparison of the solvers impedance matrix calculation time to FEKO for increasing amounts of unknowns. The MoM code is approximately a factor of 9 slower than FEKO. This is in MODE 0, the least accurate of the simulation results which is still set at a high accuracy level. Figure 3.13 shows the solvers time to calculate various sizes of impedance matrices. It takes the solver approximately 8 minutes to calculate an impedance matrix of a mesh containing ten thousand unknowns.

The MoM solver is designed to be conservative in the sense that it will rather over integrate than allow for any loss in accuracy. These runtimes can be greatly reduced if a user requires the implementation to be more efficient.

CHAPTER 3. DEVELOPMENT OF A MATLAB BASED MOM LIBRARY

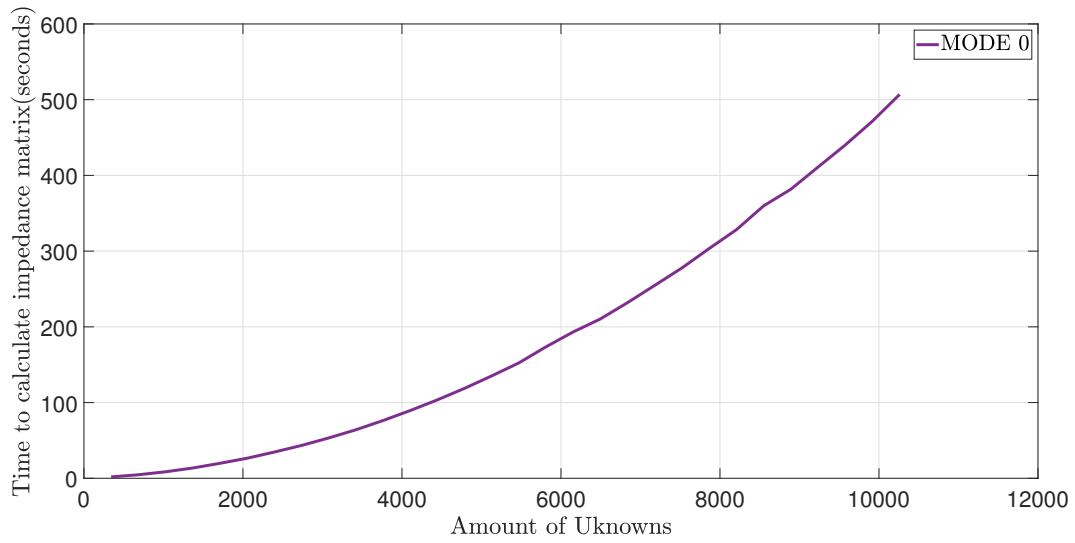


Figure 3.13: Time to calculate the impedance matrix for an increasing number of unknowns

3.4 Conclusion

The PEC EFIE RWG MoM library has now been fully described. Next an example application of the library will be presented to showcase its capabilities and benefits in providing a platform for easy development of further, complex solver capabilities.

Chapter 4

Application of MoM solver in efficient antenna array analysis

The previous chapter introduced an extensive MoM solver in which the classic MoM solution is optimized for application to a range of specialized problems. This chapter sets out to illustrate the MoM solver's utility in analysing MoM optimization techniques by employing the solver in analysis of Windowed MBF methods. They were previously introduced by the author and Serwaj K in a 2D context and are now expanded to 3D, the focus being analysis of their ability to approximate the MoM solution. It must be noted that every result and calculation in this chapter was produced using only the designed MoM solver.

4.1 Prominent MBF methods

Optimization of the MoM solution has been a primary field of study in the context of electromagnetics. Many methods of increasing the entire solutions efficiency have been well documented and studied. The ACA method takes advantage of the rank deficient nature of off diagonal sub-blocks in the MoM matrix [17], simplifying the solution of the full MoM matrix. The CBF method introduced in [4] constructs high-level expansion functions that incorporate the physics of a problem into the basis function, allowing for larger domain basis functions that result in less unknowns. The Windowed MBF method seeks to perform a similar optimization by grouping basis functions.

4.2 Linear combination of MBFs at a point

The MoM solution engine described in chapter 3 is a classic approach to solving for the surface currents on PEC structures. As discussed in the preceding section, many methods exist for reducing the MoM's computational complexity. This section introduces the concept of windowed MBFs describing the behaviour of current distributions over

CHAPTER 4. APPLICATION OF MOM SOLVER IN EFFICIENT ANTENNA ARRAY ANALYSIS

specific PEC structures. A single RWG is an approximation of the current density over a triangular pair and the linear combination of multiple RWG's describes the current behaviour over a domain consisting of N basis functions. It is recalled for convenience as,

$$\tilde{\mathbf{J}} = \sum_{n=1}^N I_n \mathbf{f}_n(\mathbf{r}) \cong \mathbf{J} \quad (4.1)$$

Say figure 4.1 represented some arbitrary current distribution over a given mesh. The linear combination of the RWG basis functions defined over this mesh, as represented by (4.1), can be considered a MBF describing the current density distribution over the mesh. It might seem intuitive that if one had an array of these geometries, and the

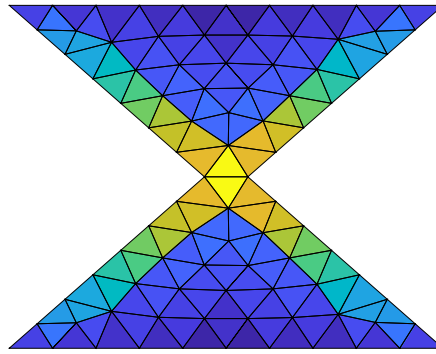


Figure 4.1: Current distribution over an arbitrary antenna

effect of their respective current distributions was desired at some far point, one could simply superimpose each of their individual effects on a point and it would accurately approximate their combined effect.

In reality, a strong inductive coupling exists between all elements in the MoM solution, this coupling is intrinsically modelled by the Z -matrix. Observing the example in figure 4.2 the bowtie antenna at position 1 is excited and the absolute values of the current distribution is taken at the middle of each triangle and interpolated. It can be observed that a current is also induced on bowties 2 and 3. Thus linearly combining the effect of the MBFs is an ineffective solution as it does not take coupling between antennas into consideration.

In order to effectively employ MBFs in obtaining the solution of scattering problems it is crucial to model their coupling effectively. The next section formally reviews the

CHAPTER 4. APPLICATION OF MOM SOLVER IN EFFICIENT ANTENNA ARRAY ANALYSIS

Windowed MBF as a means of employing MBFs to describe current distribution while including the effect of coupling between them.

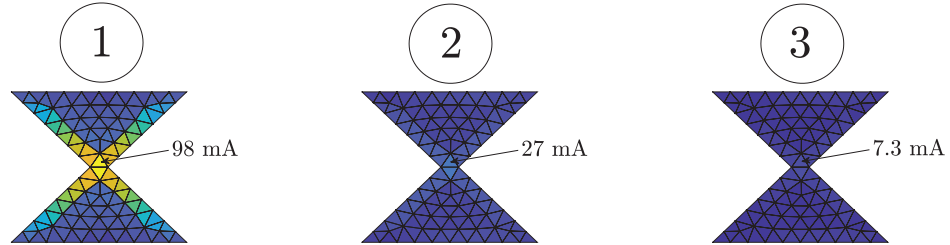


Figure 4.2: Arbitrary current distribution over an array of bowtie elements

4.3 Application of Windowed MBFs to the MoM solution procedure

Section 4.2 introduces the idea of representing current distributions over geometries as MBFs. This section formally describes how the Windowed MBF method is applied to the MoM solution procedure. The derivation and description closely follows previous work by the author. Recalling from chapter 2 the vector expression for the general MoM solution is

$$\mathbf{Z}\mathbf{I}_{MoM} = \mathbf{V} \quad (4.2)$$

\mathbf{I}_{MoM} in equation (4.2) is an $N \times 1$ vector where each entry is associated with a corresponding basis function. A Windowed MBF describes the current distribution over a certain part of a mesh and is zero everywhere else, if each MBF is defined over a unique part of the mesh and no MBFs overlap they are known as primary MBFs, excited using primary excitation. Let \mathbf{I}_{MoM} be approximated by \mathbf{I}_{MBF} as

$$\mathbf{I}_{MBF} = \mathbf{U}\boldsymbol{\beta} \quad (4.3)$$

where

$$\mathbf{U} = [\mathbf{U}_1 \quad \mathbf{U}_2 \quad \mathbf{U}_3 \quad \dots \quad \mathbf{U}_M] \quad (4.4)$$

$$\boldsymbol{\beta} = [\beta_1 \quad \beta_2 \quad \beta_3 \quad \dots \quad \beta_M]^T \quad (4.5)$$

CHAPTER 4. APPLICATION OF MOM SOLVER IN EFFICIENT ANTENNA ARRAY ANALYSIS

Observing (4.4) a single entry \mathbf{U}_m is the solution of a group of RWG's in an isolated sub-problem and β_m is the constant that scales the sub-problems solution. Substituting (4.3) into (4.2) yields

$$\mathbf{Z}\mathbf{U}\boldsymbol{\beta} = \mathbf{V} \quad (4.6)$$

observing that equation (4.6) has too many degrees of freedom both sides are multiplied by the Hermetian transpose of \mathbf{U} which gives

$$\mathbf{U}^\dagger \mathbf{Z}\mathbf{U}\boldsymbol{\beta} = \mathbf{U}^\dagger \mathbf{V} \quad (4.7)$$

equation (4.4) is a matrix that has M (The number of MBFs) columns of length N (the total number of RWG's). Each column describes the unique MBF for a specific part of the mesh. Hence \mathbf{U} will be a $N \times M$ matrix. For the primary case two things must be noted; the number of MBFs is equivalent to the number of sub meshes over which MBFs are defined, and if an arbitrary sub mesh m is chosen every entry of the corresponding m^{th} column in the \mathbf{U} vector is zero except the entries that are associated with the sub mesh. So (4.4) looks something like

$$\mathbf{U} = \begin{bmatrix} \mathbf{U}_{A1} & 0 & \dots & 0 \\ 0 & \mathbf{U}_{A2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{U}_{AM} \end{bmatrix} \quad (4.8)$$

Where each \mathbf{U}_m represents a distribution of RWG's associated with a specific surface area of a mesh structure.

The β values act as weighting coefficients that scale the MBFs. From (4.7) the values of β can be solved. If one carefully observes (4.8) the solution contains the original *mutual impedance matrix* \mathbf{Z} of the antenna array and thus the coupling effect of the impedances is taken into account here, and the approximated current \mathbf{I}_{MBF} is written

$$\mathbf{I}_{MBF} = \mathbf{U}_1\beta_1 + \mathbf{U}_2\beta_2 + \mathbf{U}_3\beta_3 \dots \mathbf{U}_m\beta_M \quad (4.9)$$

Section 4.2 highlighted that the effect of various currents on a point cannot be described by their linear combination, coupling must be taken into account. The Windowed MBF method is able to effectively model the coupling between MBFs as it takes into account the original *mutual impedance matrix*.

CHAPTER 4. APPLICATION OF MOM SOLVER IN EFFICIENT ANTENNA ARRAY ANALYSIS

Although this method would in theory yield a relatively accurate approximation of the MoM current, the MBFs are only defined over their primary geometrical domain and are zero over the rest of the geometry. Thus, the MBFs do nothing to describe the current behaviour that is induced as observed in figure 4.2. The following section describes how the Windowed MBF method can be expanded to better approximate the full MoM solution.

4.4 Enhancement of the Windowed MBF

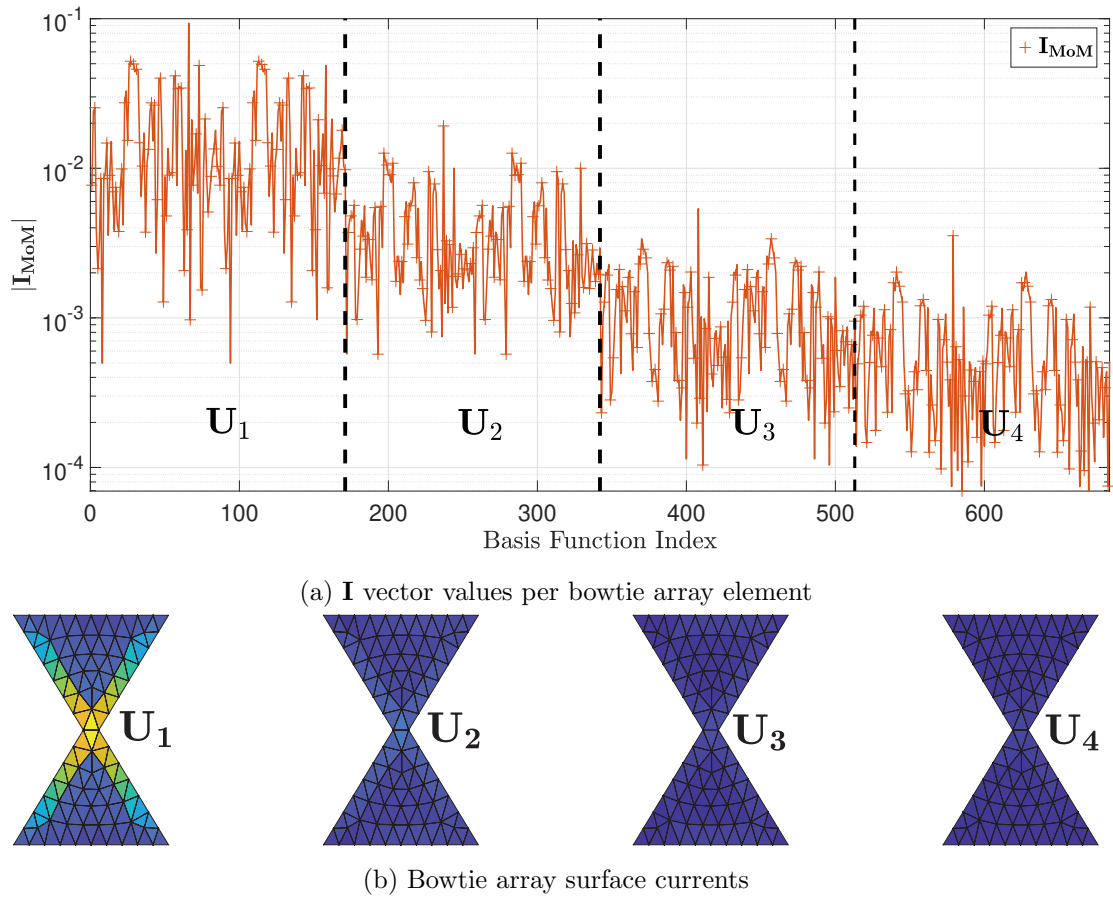
The primary Windowed MBF method is not effective enough in approximating the MoM solution and must be further enhanced. The enhancement of the Windowed MBF method is described in the subsequent sections.

The Windowed method has been assessed in a 2D context by the author of this piece as well as in [18], the findings showed it to be effective in approximating the MoM solution. The application of the Windowed method in 3D uses the exact same principles but are more challenging to implement. The intuition behind the implementation is subsequently introduced.

4.4.1 Intuitive approach

Figure 4.2 showed that exciting a single element in the array of structures induces a current over the neighbouring elements. The method described in 4.3 truncates the MBF defined on the excited array element to only exist over itself, even though it has a significant effect on its neighbours. Figure 4.3b shows a 1D array of 4 bowtie elements, where only bowtie 1 is excited at the centre. Figure 4.3a depicts the values of the current coefficients corresponding to each array element as separated by the verticle lines, current is induced over all 4 bowties. From these figures It is clear that one cannot simply truncate the basis functions as the primary windowed method suggests.

An intuitive approach to improving the Windowed method would then be to define the MBF over the excited array element as well as its neighbouring elements. \mathbf{U}_1 , \mathbf{U}_2 , \mathbf{U}_3 and \mathbf{U}_4 are defined as unique basis functions over each bowtie element that are induced by the excitation of bowtie 1. The \mathbf{U} vector from equation (4.4) would then include more MBFs and yield a better approximation of the MoM solution. In order

CHAPTER 4. APPLICATION OF MOM SOLVER IN EFFICIENT ANTENNA
ARRAY ANALYSIS

 Figure 4.3: Multiple bowtie \mathbf{I} vector illustration

to include only a finite amount of elements a radius is defined within which RWG's are to be included.

4.4.2 Defining the Windowed MBF radius

This section serves to illustrate how the Windowed method's solution radius is defined. For the sake of simplicity, only electrically disconnected structures are used in this example. An element refers to an electrically disconnected bowtie mesh in the array. Each element has a unique set of RWG basis functions defined over it.

Observing figure 4.4 the solution radius defines a set of electrically disconnected mesh elements over which MBFs will be defined. In each sub-problem only the primary antenna will be excited and MBFs are defined over each antennas included in the radius.

CHAPTER 4. APPLICATION OF MOM SOLVER IN EFFICIENT ANTENNA ARRAY ANALYSIS

The radius is depicted up to 3 which can include up to 25 basis functions. The size of each sub-problem quickly increases with the radius, hence the windowed MBF method must be employed in extremely large array problems for true benefit.

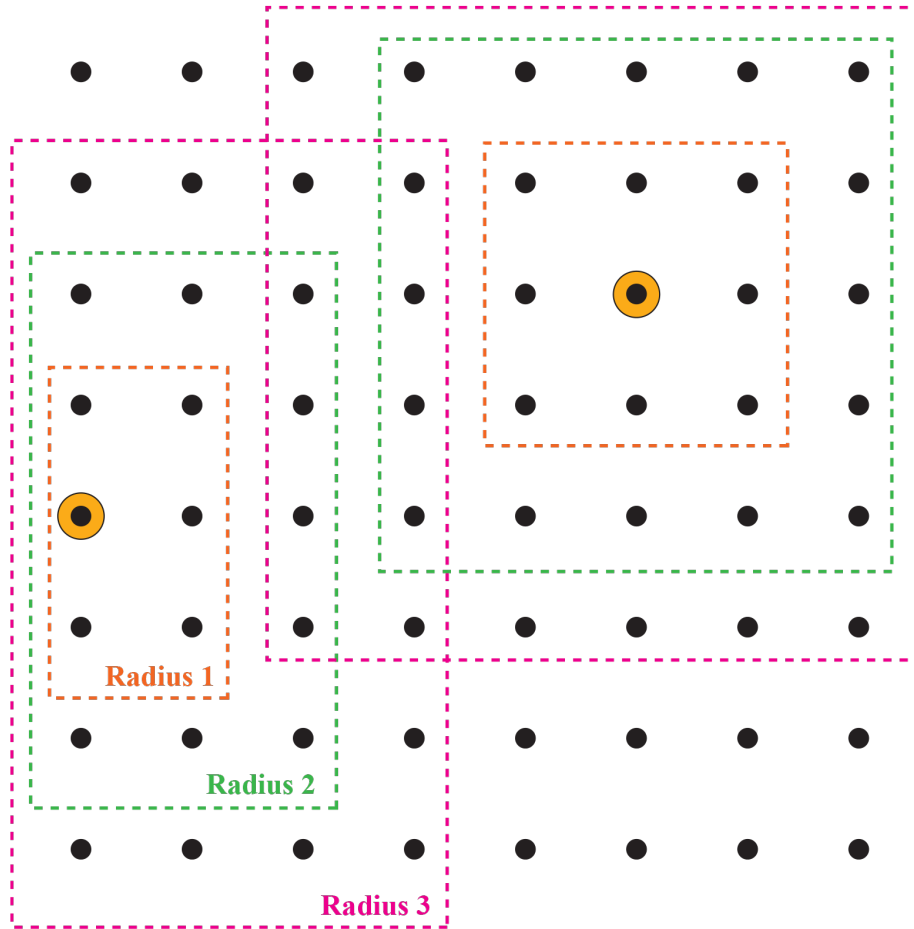


Figure 4.4: Solution radius 1, 2 and 3 illustration

4.4.3 Windowed MBF solution method

Sections 4.4.1 and ?? introduced a method of improving the Windowed MBF methods ability to approximate the MoM solution. A bounded example is now used to illustrate how the concept of a solution radius can be applied to the primary Windowed MBF method discussed in section 4.3.

An array of electrically disconnected structures, each with a finite amount of RWG's

CHAPTER 4. APPLICATION OF MOM SOLVER IN EFFICIENT ANTENNA
ARRAY ANALYSIS

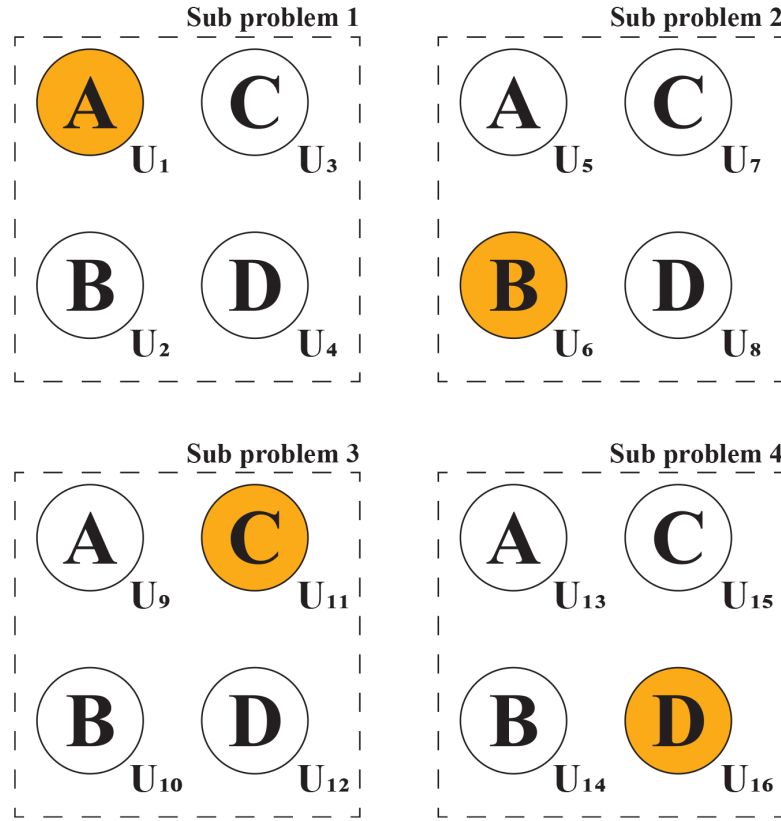


Figure 4.5: Solution radius 1 and 2 illustration

are represented in figure 4.5, the elements are labelled **A**, **B**, **C**, **D**. The solution radius is defined as 1 and each array element is excited individually to create a sub-problem. Figure 4.5 shows that each sub-problem results in four basis functions and thus the \mathbf{U} matrix will have 16 columns, note that each sub-problem does not result in a single MBF, but rather creates an MBF for each element within the solution radius. Vectors \mathbf{U} and $\boldsymbol{\beta}$ can now be written as,

$$\mathbf{U} = [\mathbf{U}_1 \quad \mathbf{U}_2 \quad \mathbf{U}_3 \quad \dots \quad \mathbf{U}_{16}] \quad (4.10)$$

$$\boldsymbol{\beta} = [\beta_1 \quad \beta_2 \quad \beta_3 \quad \dots \quad \beta_{16}]^T \quad (4.11)$$

Recalling that each MBF is specific to a certain single array element and zero on all

CHAPTER 4. APPLICATION OF MOM SOLVER IN EFFICIENT ANTENNA ARRAY ANALYSIS

other elements, the \mathbf{U} matrix can now be written more intuitively as

$$\begin{bmatrix} \mathbf{U}_1 & 0 & 0 & 0 & \mathbf{U}_5 & 0 & 0 & 0 & \mathbf{U}_9 & 0 & 0 & 0 & \mathbf{U}_{13} & 0 & 0 & 0 \\ 0 & \mathbf{U}_2 & 0 & 0 & 0 & \mathbf{U}_6 & 0 & 0 & 0 & \mathbf{U}_{10} & 0 & 0 & 0 & \mathbf{U}_{14} & 0 & 0 \\ 0 & 0 & \mathbf{U}_3 & 0 & 0 & 0 & \mathbf{U}_4 & 0 & 0 & 0 & \mathbf{U}_{11} & 0 & 0 & 0 & \mathbf{U}_{15} & 0 \\ 0 & 0 & 0 & \mathbf{U}_4 & 0 & 0 & 0 & \mathbf{U}_8 & 0 & 0 & 0 & \mathbf{U}_{12} & 0 & 0 & 0 & \mathbf{U}_{16} \end{bmatrix} \quad (4.12)$$

Figure 4.5 along with equation (4.12) illustrates the Windowed MBF method at its core. When the β coefficients are now solved for, the solution no longer only includes coupling modelled by the original Z -matrix, but also the coupling induced by array elements within the solution radius. The approximated current coefficients existing over element \mathbf{A} are written as,

$$\mathbf{I}_A = \mathbf{U}\beta \quad (4.13)$$

$$= \mathbf{U}_1\beta_1 + \mathbf{U}_5\beta_5 + \mathbf{U}_9\beta_9 + \mathbf{U}_{13}\beta_{13} \quad (4.14)$$

The expression in (4.13) includes the effect of \mathbf{A} being excited on all of the array elements, this approximation should then yield perfect reconstruction of the MoM solution. When the solution radius does not include the entire array \mathbf{I}_{MoM} will not be perfectly approximated. For larger arrays the Windowed MBF method will yield a strong approximation of the MoM current while only having to solve smaller sub-problems of the Z -matrix.

4.5 Results of applying the Windowed MBF method using the MoM solver

The Windowed MBF method has been introduced as a means of approximating the MoM solution. The MoM solver is now employed to examine the effectiveness of the Windowed MBF technique. Note that every result presented here is produced using the MoM solver. The only external step of this solution was the creation of the Bowtie mesh.

4.5.1 Primary Windowed MBFs compared to the MoM

Section 4.3 introduced the primary case of the windowed MBF method. The primary case defines MBFs over each array element and makes it zero on all other elements.

CHAPTER 4. APPLICATION OF MOM SOLVER IN EFFICIENT ANTENNA ARRAY ANALYSIS

Thus, each sub-problem created by the procedure in 4.4.3 is made up of only a single array element and is completely dependant on the original \mathbf{Z} -matrix to model coupling. If the array elements are sparsely placed, the Primary windowed method should yield a relatively good approximation of the MoM solution. Accuracy should decrease as the array elements are brought closer together, due to the coupling between elements increasing.

Figure 4.6 serves to illustrate the large extent to which coupling affects the current behaviour in the MoM solution. The figure plots the difference between \mathbf{I} vector entries associated with each basis function in the full MoM solution and \mathbf{I} vector entries in the primary windowed approximation. It plots the \mathbf{I} over a single element in a 25 element bowtie array. The Windowed method is applied to a case where the array elements are spaced $20\lambda_0$, $10\lambda_0$ and $1\lambda_0$ apart. The sparse array is a stronger approximation of the MoM solution as the array elements have less effect on each other. The dense array ($1\lambda_0$) is a weaker approximation as the strong coupling between the array elements is not effectively modelled, hence the larger difference from the MoM solution. These

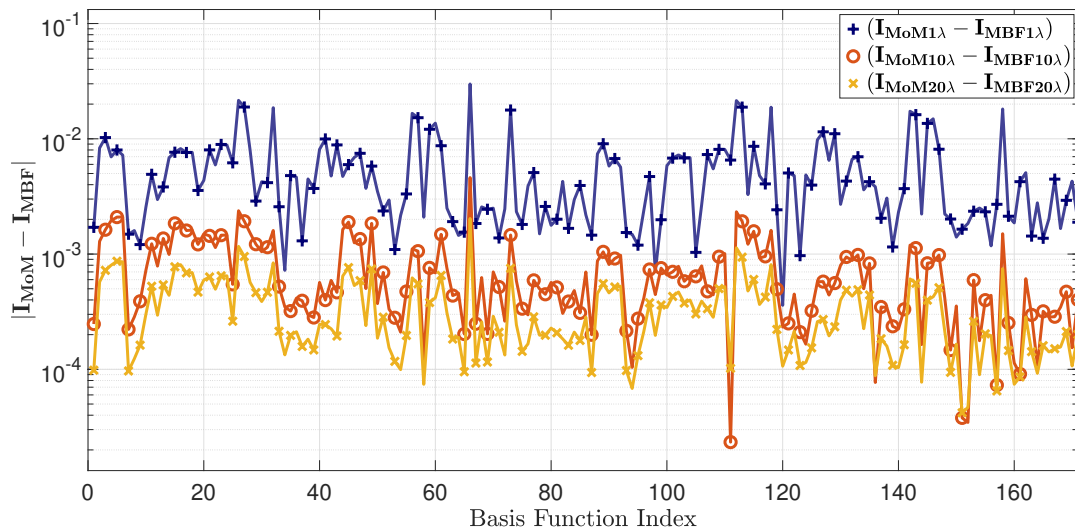


Figure 4.6: Difference of the primary case Windowed MBF approximation from the MoM at varying distances between Array elements

results show the primary Windowed MBF method is capable of approximating the MoM solution if strong coupling does not exist between the electrically disconnected array elements. The expansion discussed in 4.4 can be applied to expand the primary

CHAPTER 4. APPLICATION OF MOM SOLVER IN EFFICIENT ANTENNA ARRAY ANALYSIS

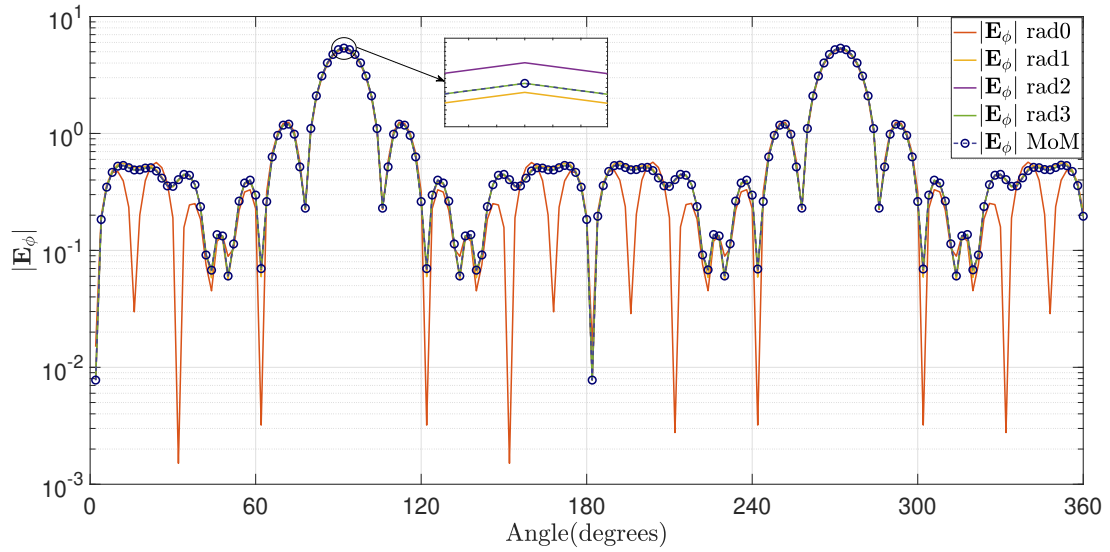


Figure 4.7: Scattered farfield of a 4x4 bowtie array in the XY plane

method to handle coupling.

4.5.2 Windowed MBFs perfect reconstruction of the MoM

The previous section showed that a single MBF defined over a unique array element could roughly approximate the MoM solution, section 4.4 discussed the method for improving this approximation by defining MBFs over multiple array elements in a single sub-problem. If the radius within which MBFs are defined is large enough that each sub-problem contains the entire mesh array, the windowed method should be capable of perfectly reconstructing the MoM solution. These results set out to prove that this is indeed possible. Figure 4.7 shows the absolute value of the E-field in the XY plane, for a 4 by 4 array of bowties. The drastic improvement in approximation of the MoM solution is evident as the radius increases beyond zero, after which the approximation steadily improves as the radius increases. Figure 4.8 plots the differences of increasing radius MoM approximations from the MoM solution. When the radius reaches 3 each sub-problem contains all of the bowties and radius 3's difference drops to a value that can be allotted to machine error. Thus, the Windowed MBF method can perfectly recreate the MoM, this proves that the method is viable in a 3D context.

CHAPTER 4. APPLICATION OF MOM SOLVER IN EFFICIENT ANTENNA ARRAY ANALYSIS

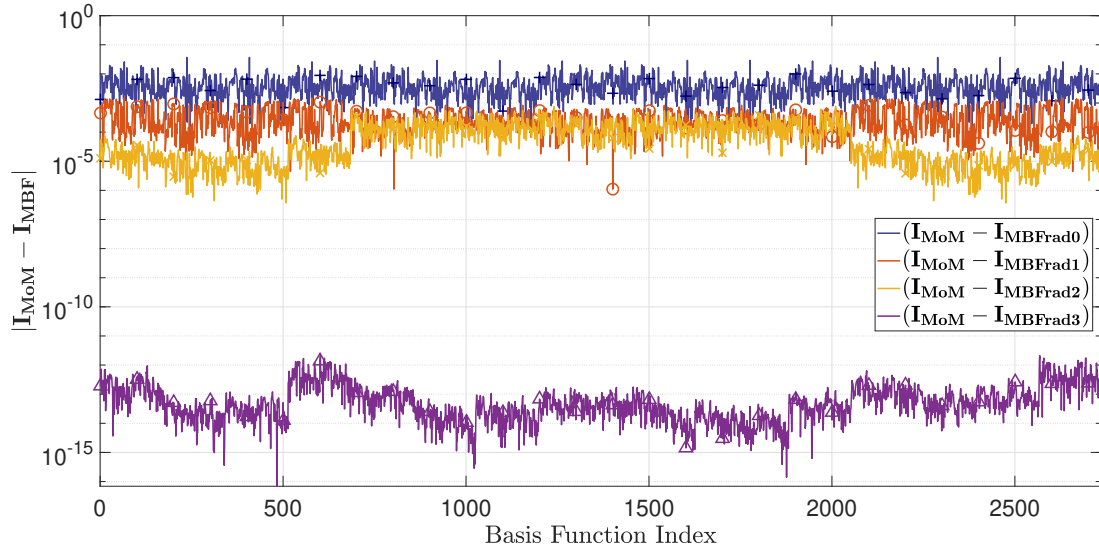


Figure 4.8: Difference of varying radius Widowed MBF current vector solutions from MoM solution

4.5.3 Windowed MBFs applied to a large scale problem

Section 4.5.2 showed that the MoM solution could be approximated using the Windowed MBF method, but to achieve a perfect MoM reconstruction the radius must include every array element. This section applies the Windowed MBF method to a large array of 64 bowties, assessing how well it converges and whether it is a viable approximation of the MoM solution. The scattered field and current vector are plotted for various radius' of a 64 bowtie array in figures 4.9, 4.10 and 4.11. All of these results show a consistently improving approximation of the MoM as the solution radius increases.

4.5.4 Conclusion

This chapter set out to test the viability of the Windowed MBF method as well as illustrating the developed MoM solver's effectiveness in custom applications. The windowed MBF method is most definitely capable of accurate approximations of the MoM solution and the approximation converges as solution radius is increased.

CHAPTER 4. APPLICATION OF MOM SOLVER IN EFFICIENT ANTENNA ARRAY ANALYSIS

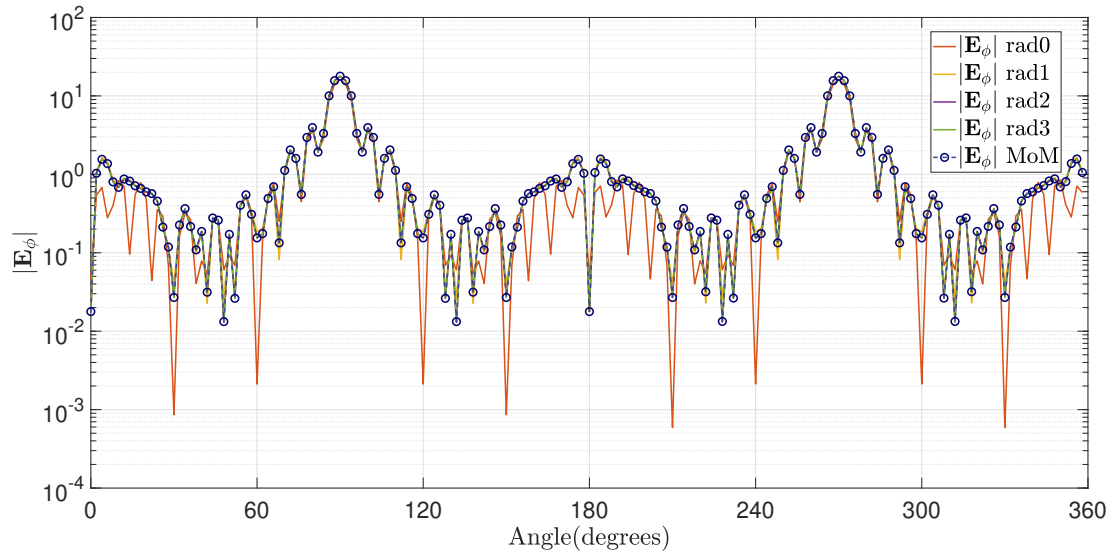


Figure 4.9: 64 Bowtie scattered field in XY plane

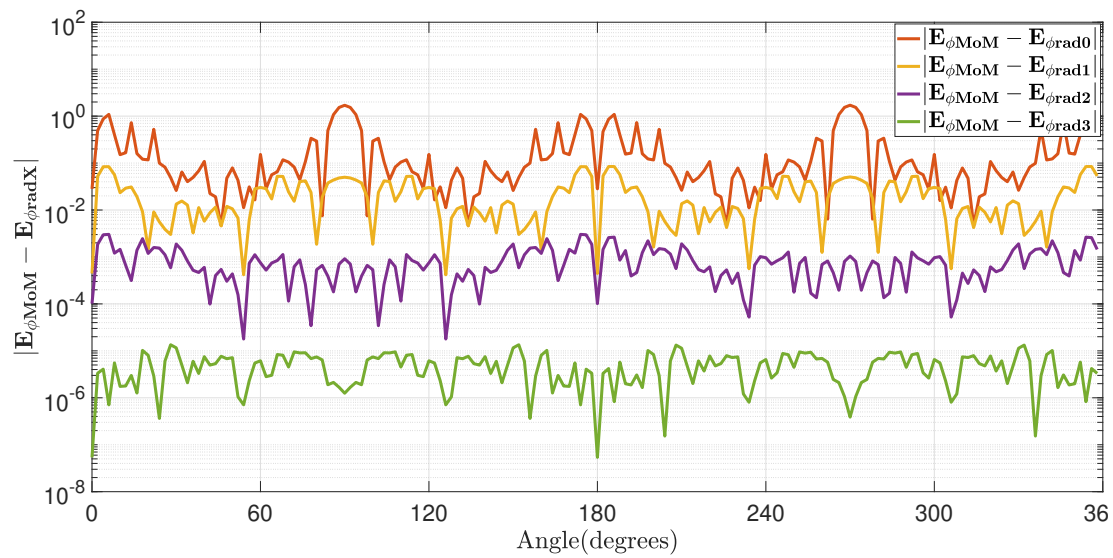


Figure 4.10: 64 Bowtie scattered field in XY plane difference from MoM solution

CHAPTER 4. APPLICATION OF MOM SOLVER IN EFFICIENT ANTENNA ARRAY ANALYSIS

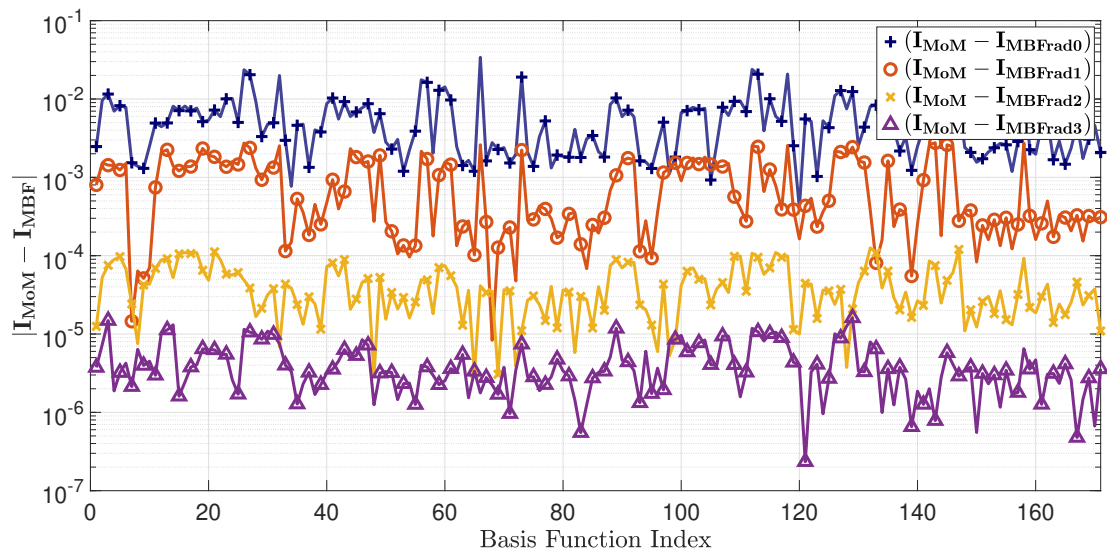


Figure 4.11: Difference of varying radius Widowed MBF \mathbf{I} vector solutions from full MoM solution of 64 bowties

Chapter 5

Conclusions and recommendations

A highly accurate, efficient and easily customisable RWG MoM solver was developed in this thesis and was showcased by the implementation and analysis of Windowed MBFs in a 3D context. For emphasis the articles published based on this code are again listed

- *K. Serwaj, M. M. Botha "Computation of MBF Reaction Matrices for Antenna Array Analysis, with a Directional Method", [15]*
- *M. Chose, M.M. Botha "Improvements to the Domain Green's Function Method for Antenna Array Analysis", [16]*
- *M. Chose, M.M. Botha "An iterative method for the analysis of large disjointed antenna arrays", submitted for EUCAP 2020*
- *K. Serwaj, M. M. Botha "Directional Method to Compute Reduced Matrix System in MBF Solvers", submitted for EUCAP 2020*

The EFIE-based RWG MoM solver was developed for extreme accuracy. Custom Gaussian quadrature and Radial-Angular-R-Sqrt functions were developed to handle standard integration and singularity cancellation integration respectively. Assigning the amount of points used for integration was done dynamically to ensure maximum accuracy without over integration. After the accurate solver was developed it was expanded to cater for the request of specific basis function solutions. Once all the required functionality was added to the solver it was optimized massively. A MoM toolkit was then developed for custom applications. Finally the MoM solver and toolkit were employed to develop Windowed MBFs in a 3D context as a showcase of the code and to test whether windowed MBFs are accurate in a 3D context.

Recommendations and future work This code is specifically designed for expansion and improvements. I recommend the code be slightly more formalized and loaded onto a git repository for public use. Making this code easily accessible for many students and researchers is highly beneficial to all parties involved, it encourages collaboration

CHAPTER 5. CONCLUSIONS AND RECOMMENDATIONS

and feedback. Although the RWG MoM solution is still widely used, a higher order basis function method could be applied to the code for further improvements in simulation results. For running on a standard desktop or laptop, parallelization should be considered for speed improvements. The code is extremely conservative when assigning integration points and it often assigns too many integration points in many regions. It can be improved by making the logic more responsive to triangle size and orientation, this will further reduce runtime.

References

- [1] J.-M. Jin, *The finite element method in electromagnetics*. New York: Wiley, 2nd ed. ed., 2002. [1](#), [21](#)
- [2] A. F. Peterson, *Computational methods for electromagnetics*. IEEE/OUP series on electromagnetic wave theory, Piscataway, N.J.: IEEE Press, 1998. [1](#)
- [3] S. Kurz, O. Rain, and S. Rjasanow, “The adaptive cross-approximation technique for the 3d boundary-element method,” *IEEE Transactions on Magnetics*, vol. 38, no. 2, pp. 421–424, 2002. [2](#)
- [4] V. V. S. Prakash and R. Mittra, “Characteristic basis function method: A new technique for efficient solution of method of moments matrix equations,” *Microwave and Optical Technology Letters*, vol. 36, no. 2, pp. 95–100, 2003. [2](#), [45](#)
- [5] R. Maaskant, R. Mittra, and A. Tijhuis, “Fast Analysis of Large Antenna Arrays Using the Characteristic Basis Function Method and the Adaptive Cross Approximation Algorithm,” *IEEE Transactions on Antennas and Propagation*, vol. 56, no. 11, pp. 3440–3451, 2008. [2](#)
- [6] J.-M. Jin., *Theory and Computation of Electromagnetic Fields*. New York: John Wiley and Sons, 2010. [4](#), [10](#), [11](#)
- [7] C. A. Balanis, *Advanced engineering electromagnetics*. New York, N.Y.: Wiley, 1989. [4](#), [5](#), [6](#), [8](#), [9](#), [14](#), [15](#)
- [8] W. C. Gibson, *The Method of Moments in Electromagnetics*. Chapman & Hall, 2008. [5](#), [7](#), [16](#)
- [9] R. F. Harrington, *Field Computation by Moment Methods*. Wiley-IEEE Press, 1993. [6](#)
- [10] J. M. Melenk and I. Babuška, “The partition of unity finite element method: basic theory and applications,” *Computer methods in applied mechanics and engineering*, vol. 139, no. 1-4, pp. 289–314, 1996. [11](#)

REFERENCES

- [11] S. M. Rao, D. R. Wilton, and A. W. Glisson, “Electromagnetic Scattering by Surfaces of Arbitrary Shape,” *IEEE Transactions on Antennas and Propagation*, vol. 30, no. 3, pp. 409–418, 1982. [12](#), [13](#)
- [12] D. A. Dunavant, “High degree efficient symmetrical gaussian quadrature rules for the triangle,” 1985. [18](#)
- [13] S. Järvenpää, M. Taskinen, and P. Ylä-Oijala, “Singularity extraction technique for integral equation methods with higher order basis functions on plane triangles and tetrahedra,” 2003. [18](#)
- [14] M. Khayat, D. Wilton, and P. Fink, “An improved transformation and optimized sampling scheme for the numerical evaluation of singular and near-singular potentials,” *IEEE Antennas and Wireless Propagation Letters*, vol. 7, pp. 377–380, 2008. [18](#), [36](#)
- [15] K. Sewraj and M. M. Botha, “Computation of mbf reaction matrices for antenna array analysis, with a directional method,” pp. 0385–0389, Sep. 2019. [20](#), [59](#)
- [16] M. Chose and M. M. Botha, “Improvements to the domain green’s function method for antenna array analysis,” pp. 0392–0395, Sep. 2019. [20](#), [59](#)
- [17] K. Zhao, M. N. Vouvakis, and J. F. Lee, “The adaptive cross approximation algorithm for accelerated method of moments computations of EMC problems,” *IEEE Transactions on Electromagnetic Compatibility*, vol. 47, no. 4, pp. 763–773, 2005. [45](#)
- [18] K. Sewraj, R. C. Beswick, and M. M. Botha, “Efficiency of Various Macro Basis Function Schemes for Antenna Array Analysis,” *Proceedings of the 2018 20th International Conference on Electromagnetics in Advanced Applications, ICEAA 2018*, pp. 254–257, 2018. [49](#)